Orbital Debris Tracking using a Wireless Sensor Network

Undergraduate Thesis

Presented in Partial Fulfillment of the Requirements for
Graduation with Research Distinction in the
Department of Mechanical and Aerospace Engineering at
The Ohio State University

by

Wilson Enrique Flores

May 2018

Advisor: Mrinal Kumar, Ph.D.

This page intentionally left blank.

ABSTRACT

One of the biggest threats faced with space exploration is a problem known as Space Situational Awareness (SSA). SSA is the ability to view, understand, and predict the physical location of space debris with the objective of avoiding collisions. Space debris, or orbital debris, is the collection of defunct human-made objects orbiting the Earth. Currently, the U.S. Space Surveillance Network is capable of tracking millions of objects larger than 1 millimeter in size using 20 sensor sites around the world. Moving at thousands of miles per hour, orbital debris is more probable to have high-speed impacts between objects. Randomly distributed around Earth's orbit, objects that do collide cause even more fragmentation debris, allowing the problem to grow. To minimize debris generation in Low-Earth orbit, monitoring with high precision is required. Although capabilities in monitoring debris already exist, being able to track smaller debris in the space environment will be important to future spacecraft design. The challenge that arises with orbital debris is predicting current and future debris trajectories. This investigation focuses on tracking a moving target using wireless sensors, in order to characterize uncertainty on the target's physical location. This study will experimentally characterize state uncertainty using robots in connection with a network of sensors within a static environment in the laboratory. Results show that uncertainty with sensors, motors, and target dynamics are all areas that can be examined further to improve target tracking. This study will provide insight to not only space applications, but also areas in wildfire monitoring.

ACKNOWLEDGMENTS

I would first like to thank my research advisor, Dr. Mrinal Kumar, for supporting me during these past two years. Thank you to his motivation, enthusiasm, immense knowledge, guidance, and patience, I was able to complete my undergraduate research thesis. I could not have imagined having a better advisor and mentor for my undergraduate study. Having the opportunity to work in the Laboratory for Autonomy in Data-Driven and Complex Systems has allowed me to learn so much about aerospace applications. Dr. Kumar has provided insightful discussions about research, which has allowed me to continue to pursue a Ph.D. in Aerospace Engineering. Thank you Dr. Kumar for your continuous support.

I would also like to thank the other professor in my undergraduate thesis committee, Dr. John Horack, for all of his support in my undergraduate career here at Ohio State. Thank you Dr. Horack for your constant guidance and motivation. You have truly allowed me to fall in love even more with Aerospace Engineering. Your love for teaching is contagious and I hope to one-day deliver amazing lectures just as you did. Thank you Dr. Horack for your continuous support.

I would also like to thank my graduate student mentor, Mr. Alex Soderlund, for his continued guidance and advice throughout the entire research development and analysis phase. I could not have done any of this without your help. Thank you Alex for your continuous support.

I especially want to thank my family for their continuous love and support through my college career. Thank you mom and dad for being the hardest-working parents I know. Thank you for the sacrifices you made to allow me to be the first to attend college. I love them so much, and I would not have made it this far without them. Thank you to my two brothers for their support in watching me accomplish my dreams. They will forever be my lifelong best friends. Special thanks

to my abuelita for her loving words and always believing in me. I know if she were still here, I would be making her so proud.

Finally, I would like to thank my beautiful fiancé for her continuous love and support throughout college. You have been my inspiration, motivation, and greatest supporter of all that I do. Thank you for you unconditional love during my good and bad times. I could have not been as successful as I am if it was not for you. Thank you Yeliani Valdez for never giving up on me. I promise to love you every day, forever.

Overall, I would like to thank God Almighty for giving me the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily. Without his blessings, this achievement would not have been possible.

I dedicate this undergraduate thesis to
my parents, my brothers, and my beloved fiancé, Yeliani,
for their constant support and unconditional love.
I love you all dearly.

TABLE OF CONTENTS

ABSTRACT iii
ACKNOWLEDGMENTS iv
TABLE OF CONTENTS vi
LIST OF FIGURES viii
LIST OF TABLES ix
CHAPTER 1 – INTRODUCTION 1
1.1 Orbital Debris 1
1.2 Space Situational Awareness 3
1.3 Space Surveillance Network 5
1.4 Space Debris Target Tracking 6
1.5 Focus of Research 8
1.6 Significance of Research 8
1.7 Overview of Thesis 9
CHAPTER 2 – METHODOLOGY 9
2.1 TurtleBot 9
2.2 Kinect Camera 10
2.3 Robot Operating System 11
2.4 Wireless Sensor Network 12
2.5 Research Approach 12
2.6 Data Collection Method and Tools 12
2.7 Experimental Test Setup 13
2.8 Research Process 20
2.9 Data Analysis 21
2.10 Extended Kalman Filter 21
2.11 Research Limitations 22
CHAPTER 3 – RESULTS 22
3.1 Overview 22
3.2 Range Calibration Test 23
3.3 Bearing Calibration Test 25
3.4 Circular Orbit Calibration Test 28
3.5 Basic Space Situational Awareness Problem Test 31
3.6 Orbit Characterization using Extended Kalman Filter Test 33

CHAPTER 4 – CONCLUSION 41
4.1 Contributions 41
4.2 Additional Applications 41
4.3 Future Work 42
4.4 Summary 43
APPENDICIES 45
Appendix A 45
Appendix B 47
Appendix C 48
REFERENCES 49

LIST OF FIGURES

Figure 1: Classes of space objects [4]. 2
Figure 3: The Growth of the Satellite Population [11]. 5
Figure 4: Sensor Sites in the Space Surveillance Network [12]. 6
Figure 5: Haystack and HAX radars in Tyngsboro, MA. [16]. 7
Figure 6: Michigan Orbital Debris Survey Telescope (MODEST) in La Serena, Chile [17]. 8
Figure 7: TurtleBot™ robot [19]. 10
Figure 8: Kinect™ Sensor Schematic [20]. 11
Figure 9: TurtleBot™ Range Calibration Test Diagram 16
Figure 10: TurtleBot™ Bearing Calibration Test Diagram 17
Figure 11: TurtleBot™ Circular Orbit Calibration Test Diagram 18
Figure 12: Orbit Tracking using multiple Sensors Diagram 19
Figure 13: Orbit Characterization Test Diagram 20
Figure 14: TurtleBot™ Range Calibration Test 1 24
Figure 15: TurtleBot™ Range Calibration Test 2 25
Figure 16: TurtleBot™ Bearing Calibration Test 26
Figure 17: TurtleBot™ Bearing Calibration Test Deviations 28
Figure 18: TurtleBot™ Circular Orbit Calibration Test 29
Figure 19: TurtleBot™ Circular Orbit Calibration Test Deviations 30
Figure 20: Space Situational Awareness Problem using three TurtleBots™ 32
Figure 21: Measured Position using EKF 34
Figure 22: Mean of Measured Position using EKF 35
Figure 23: State of moving object using EKF 36
Figure 24: Mean State of moving object using EKF 37
Figure 25: Prediction of moving object using EKF 38
Figure 26: Mean Prediction of moving object using EKF 39
Figure 27: Mean Posterior, Prior, and Measured data of moving object using EKF 40

LIST OF TABLES

Table 1: Hardware List in a TurtleBot™ [18]. 10

CHAPTER 1 – INTRODUCTION

1.1 Orbital Debris

As humanity continues to push forward with space exploration, a challenge arises with space objects traveling around Earth at thousands of miles per hour. These high-speed space objects include both natural and manmade objects orbiting the Earth. Some of these objects do fall back into Earth's atmosphere, allowing them to either burn up and disintegrate on reentry, or fall onto the surface of the Earth, mostly the oceans [1]. Most objects remain in orbit for long periods of time, which has caused thousands of space objects to clutter the space environment around the Earth. In over the past 50 years of human space exploration, the collection of space debris, or orbital debris, found in the space environment around the Earth has only become worse. Space debris, also known as space junk, is any manmade object in orbit about the Earth, which no longer serves a useful function [2]. Currently right now, scientific models estimate that there are more 170 million space debris objects larger than 1 mm, 670,000 larger than 1 cm, and 29,000 larger than 10 cm in orbit [3]. As more missions to space are undertaken, these numbers will continue to grow. Figure 1 depicts the range of objects found in space, including various types of debris [4].

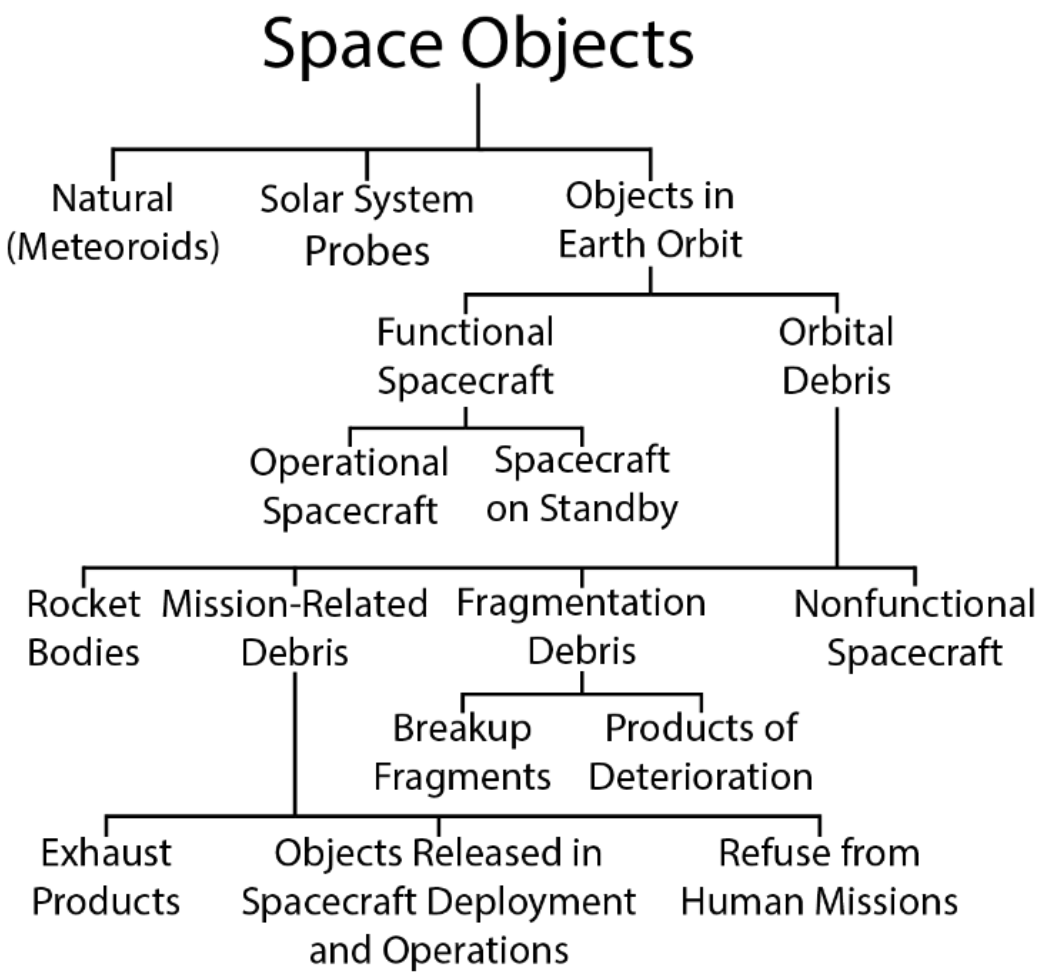

Figure 1: Classes of space objects [4].

With such a large amount of orbital debris found around the Earth, a challenge arises with space collisions. Orbital debris becomes a problem to functional assets, creating a risk of an accidental hypervelocity collision. The first accidental hypervelocity between to intact spacecraft occurred on February 10, 2009. Iridium 33, a U.S. operational communications satellite, and Cosmos 2251, a Russian decommissioned communications satellite, collided at nearly right angles from each other's orbital planes [5]. This collision then produced a significant debris cloud allowing the number of orbital debris to increase.

Orbital debris has also caused hazard to humans at the International Space Station (ISS). A crew onboard the ISS had to temporarily retreat to a safety zone on March 12, 2009, when a

small piece of orbital debris was estimated to come close to the ISS [5]. Passing at a safe distance away, the orbital debris caused no damage to the ISS. A delayed recognition of orbital debris can create threat to both space assets and humans, and may leave no time to prepare for a collision avoidance maneuver [5].

As years go by, the total number of objects in Earth orbit will continue to grow. As seen in Figure 2, the object type that has the largest number in orbit is fragmentation debris. This debris therefore, has to be searched for, discovered, tracked, and maintained in a database to obtain a high accuracy space object catalog.

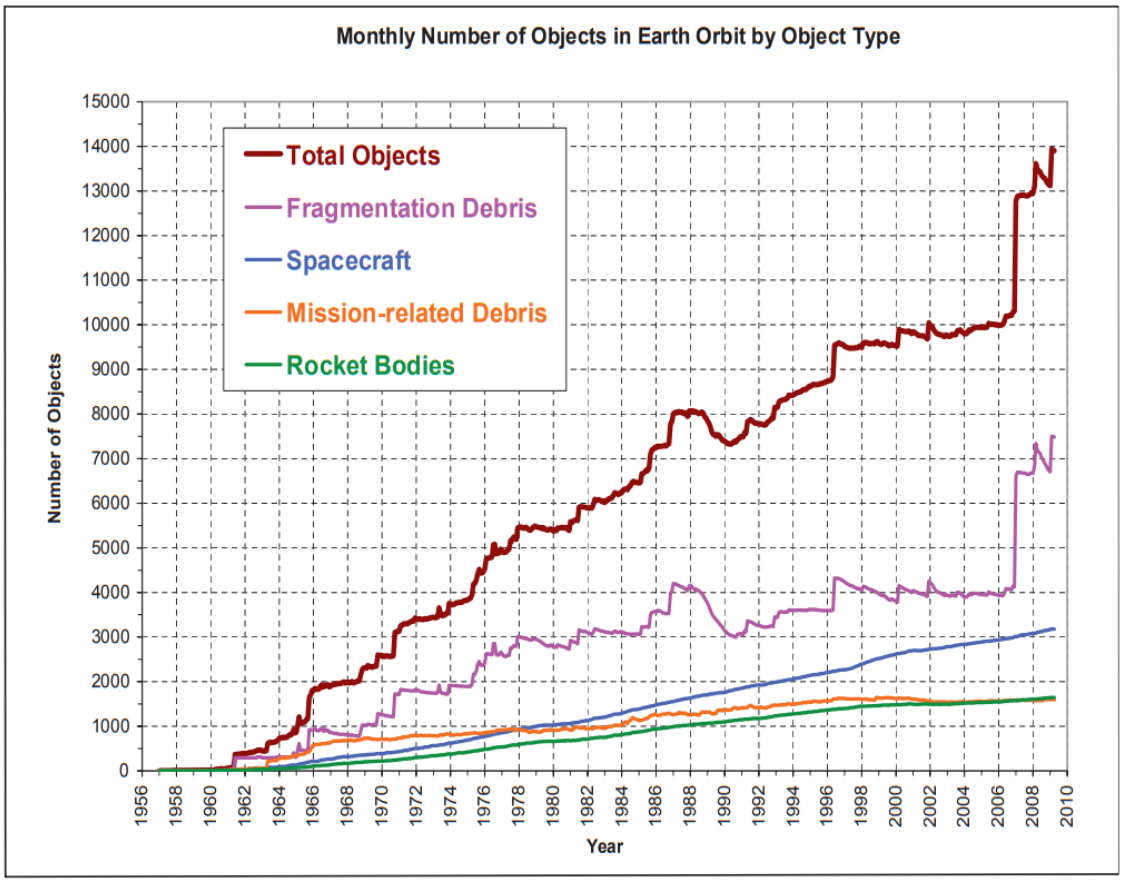

Figure 2: Monthly Number of Cataloged Objects in Earth Orbit by Orbit Type [5].

1.2 Space Situational Awareness

"Space Situational Awareness (SSA) refers to the ability to view, understand, and predict the physical location of natural and manmade objects in orbit around the Earth, with the objective

of avoiding collision," as stated by the Space Foundation [6]. The U.S. Strategic Command defines SSA as "the requisite current and predictive knowledge of space events, threats, activities, conditions, and space system (space, ground, link) status capabilities, constraints and employment – to current and future, friendly and hostile – to enable commanders, decision makers, planners, and operators to gain and maintain space superiority across the spectrum of conflict [7]." In a broad perspective, SSA is the knowledge of our near-space environment, including artificial and natural objects.

SSA is a term that has become prominent recently due to several collisions of orbiting space objects [8]. SSA aims to enable us to autonomously detect, predict, and assess risk due to manmade space debris objects [9]. The growth of the satellite population is increasing as years go by, as seen in Figure 3. This means that current artificial space debris tracking methods need improvement to better understand the future effects this challenge may cause. SSA is a necessity for any nation that seriously bases its military and economic well-being even partly on space capabilities [10]. By improving SSA, accidental collisions in the future can be avoided allowing space assets to remain safe in orbit.

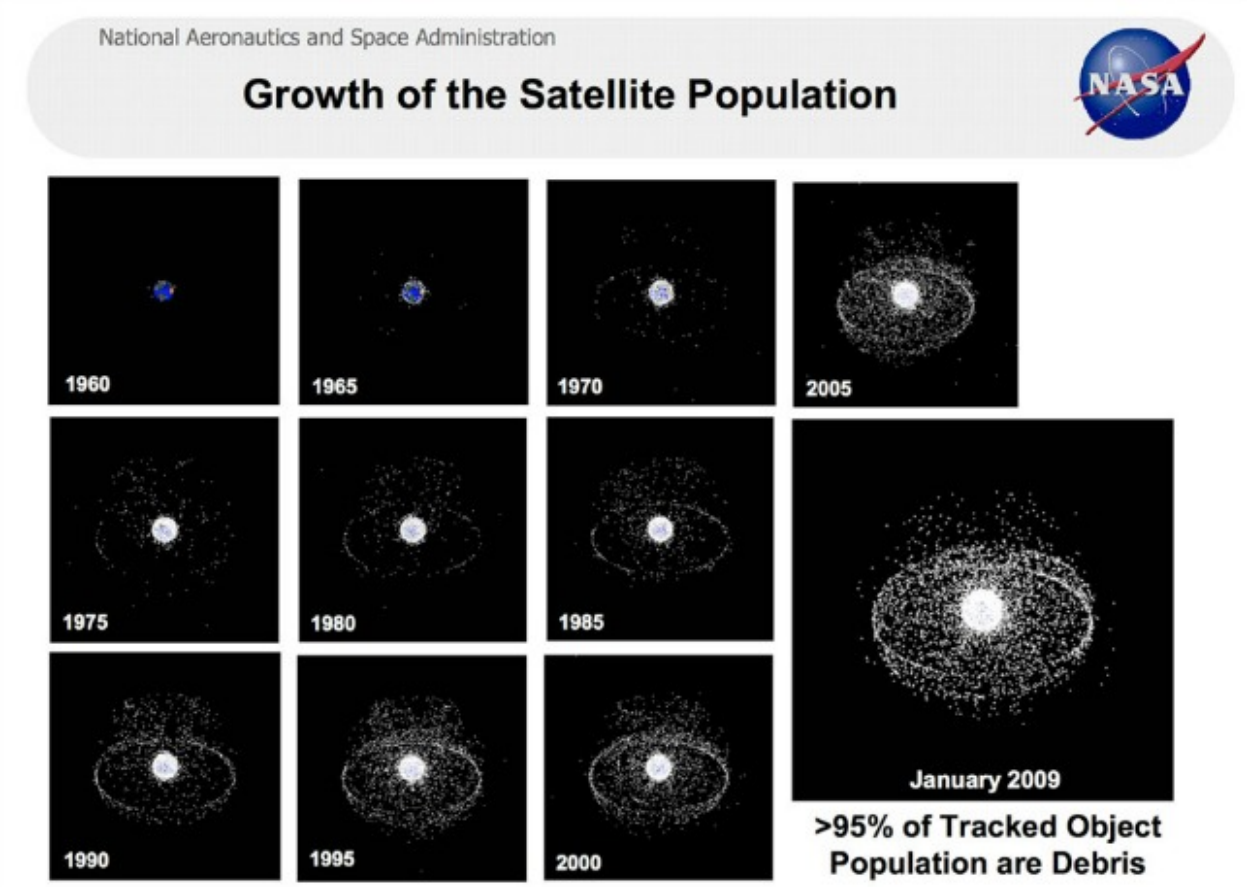

Figure 3: The Growth of the Satellite Population [11].

1.3 Space Surveillance Network

To minimize the threat of space collisions between spacecraft and artificial space debris, the U.S. Space Surveillance Network (SSN) tracks orbital debris using a network of 20 optical and radar sensor sites located across the globe. Operated by the U.S. Department of Defense, the SSN helps predict and alert satellite operators, in particular U.S. allies, when a debris-creating event has occurred [12]. This allows satellite operators to perform collision avoidance maneuvers, potentially prolonging the satellite's life and revenue. However, because of the large number of orbital debris and the limited number of sensors available (see Figure 4) to track these objects, it is impossible to maintain persistent surveillance on all objects orbiting the Earth [13].

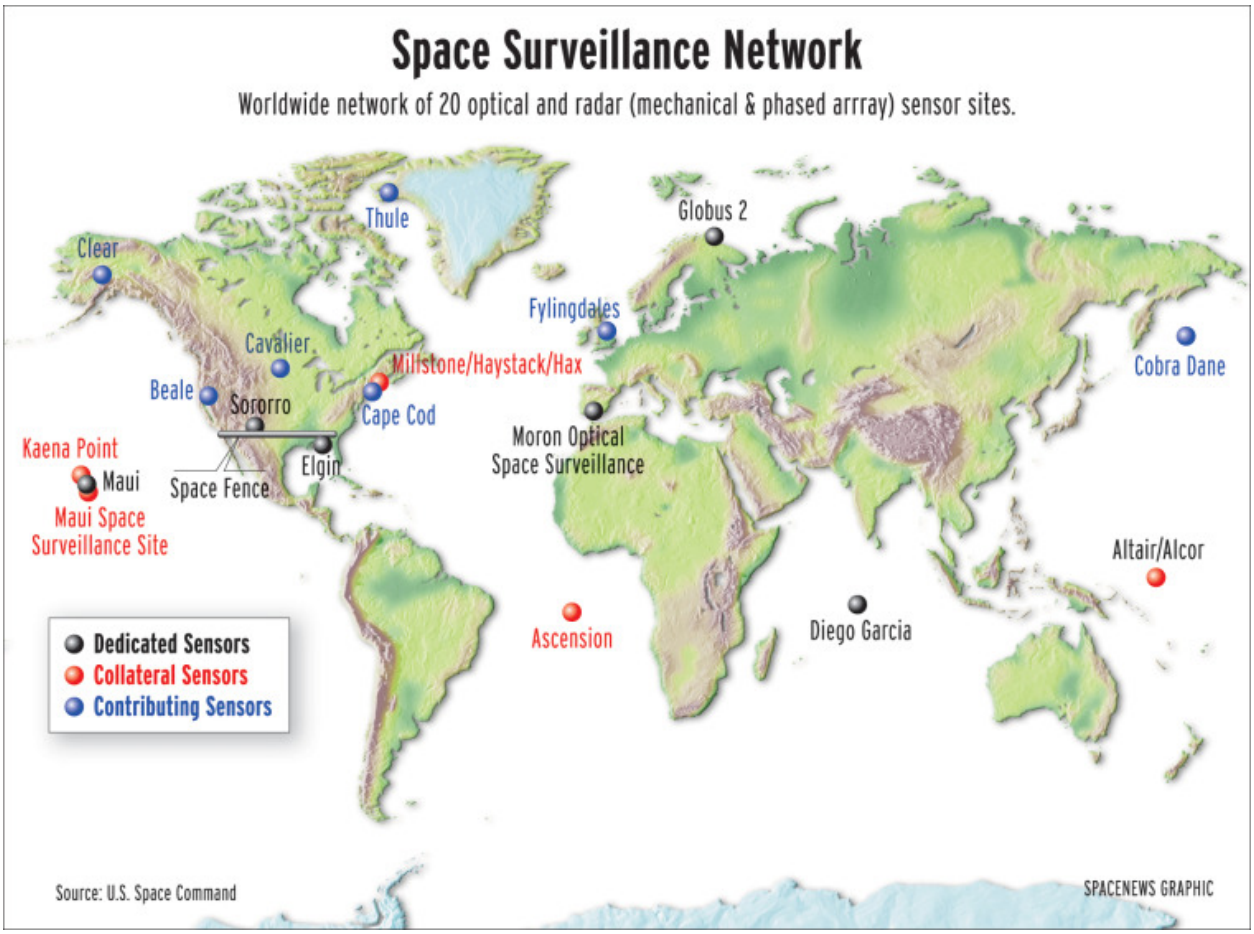

Figure 4: Sensor Sites in the Space Surveillance Network [12].

1.4 Space Debris Target Tracking

To be able to maintain knowledge of the space environment, SSA requires the knowledge of the current and predicted states of space objects [14]. SSA currently tracks thousands of objects via limited ground-based sensors that include radars and telescopes (see Figures 5-6 for examples), to be able to estimate and predict the location of space objects. By using this method, a space object catalog was created to store space object information, including object characterization. To maintain the current and predicted states of space objects, the Joint Space Operations Center (JSpOC) collects and processes 400,000 observations every day to update the estimated states of over 20,000 objects [15]. Although this method of space object tracking has worked in the past, improved methods of multi-target tracking for space objects could provide a potential means to

reduce uncertainty and improve orbit state estimates [14]. The U.S. Air Force has provided recommendations to reevaluate the current methods used for target tracking. They have suggested improving data association methods, improving estimation algorithms, and improving the handling of large quantities of data collection. Space debris tracking poses a unique challenge with opportunities for improvement.

Figure 5: Haystack and HAX radars in Tyngsboro, MA. [16].

Figure 6: Michigan Orbital Debris Survey Telescope (MODEST) in La Serena, Chile [17].

1.5 Focus of Research

The focus of my research involves a simplified SSA problem in which I track a moving target in a circular pattern. The goal of the research is to try to recreate a SSN that will recognize a target moving around in a circular orbit-shaped pattern and predict its location in the future. I solely focused on the experimental side of this research, working with a team of robots to recreate orbits and collect sensor data. Understanding the capabilities and limitations these robots have was my focus to help further research prove better methods on tracking orbital debris.

1.6 Significance of Research

As space exploration continues forward, more rockets and satellites will be launched into space, leaving behind possible debris from upper stage explosions, tiny flecks of paint released by thermal stress, or solid rocket motor effluents. This debris in orbit has now started to accumulate, and will continue to grow if nothing takes action now. The significance of this research is simple, space sustainability. If humans want to continue visiting the ISS, or add more powerful

communication satellites in orbit, orbital debris will remain a threat to those actions if space sustainability solutions are not undertaken. This research will not only allow us to be able to understand where this space debris is now, but it will allow us to predict where they will be in the future, which could help us find solutions to space sustainability.

This research could also help the world continue improving future predictions of potential collisions between spacecraft and orbital debris. These predictions could help operators obtain timely and actionable information, allowing collision avoidance through satellite maneuvers. Avoidance of collisions would not only protect government assets, but also allow less debris to form from a hypervelocity impact.

1.7 Overview of Thesis

This thesis is outlined as follows. The next chapter includes methods used in this research. The experimental methodology will be discussed in detail, as well as experimental equipment used for this research. Chapter 3 will present and discuss all of the experimental findings from this research. These results will include calibration data, target tracking data, orbit determination, and Extended Kalman Filter observations. Finally, Chapter 4 will discuss the research conclusions and recommendations for future research.

CHAPTER 2 – METHODOLOGY

2.1 TurtleBot

For this specific research, the most important piece of equipment used to conduct experiments in the laboratory was a TurtleBot™ robot. A TurtleBot™ robot (see Figure 7) is a robot kit that provides hardware and software for development of robot algorithms [18]. The hardware consists of an iRobotCreate, a Kinect™ camera, and a 150 degrees/second single axis gyro [18]. A linux open-source operating system computer was mounted on the top module plate

of the robot. This computer, known as the robot computer, is what controlled and operated the TurtleBot™. A complete list of hardware specifications in a TurtleBot™ can be found in Table 1.

Table 1: Hardware List in a TurtleBot™ [18].

Mobile Base and Power Board	3D Sensor	Computing: Lenovo Thinkpad	TurtleBot Hardware
iRobotCreate	Microsoft Kinect™	Processor: 7th Gen Intel® Core™ i7	Kinect™ mounting hardware
3000 mAh Ni-Mh battery pack	Kinect™ power board adapter cable	Memory: 32 GB DDR4	TurtleBot structure
150 degrees/second single axis gyro		Graphics: Intel® HD Graphics 620	TurtleBot module plate with 1 inch spacing hole pattern
12 V 1.5 A software enabled power supply (for powering the Kinect™)		Internal Hard Drive: 500GB 7200 RPM	

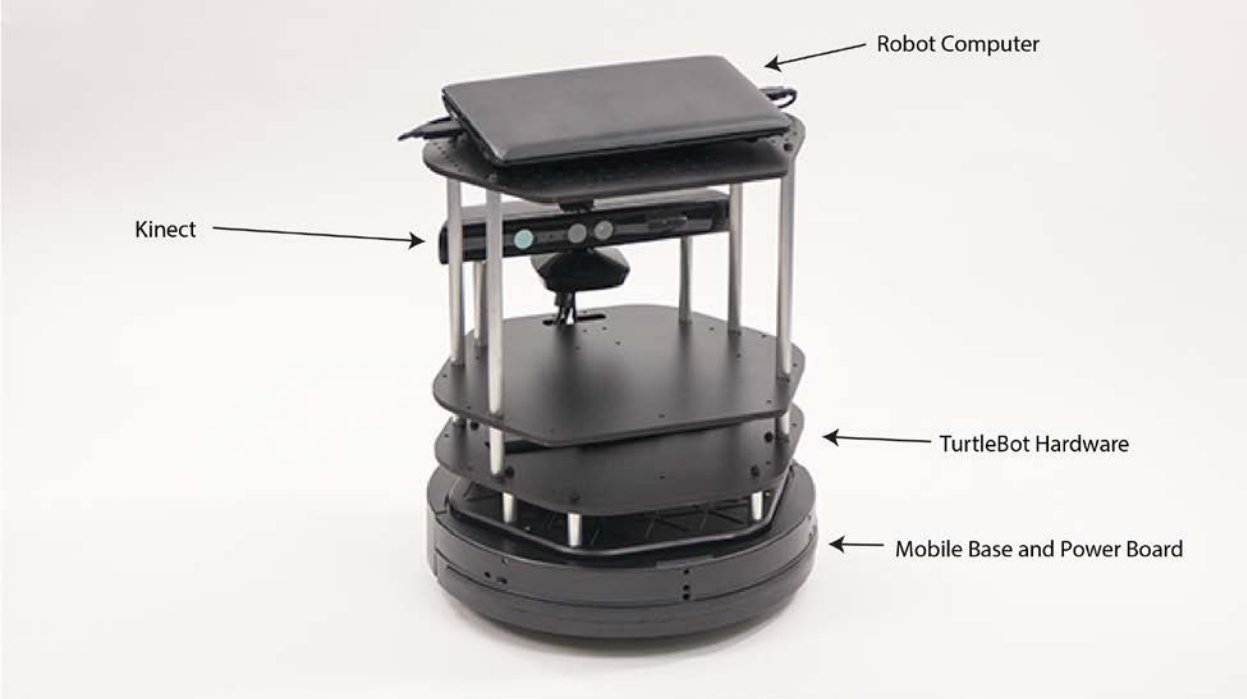

Figure 7: TurtleBot™ robot [19].

2.2 Kinect Camera

The Microsoft Kinect™ camera is a sensor specifically built for Xbox console applications, but comes integrated with the TurtleBot™ robot kit. This sensor has two cameras, one RGB-camera and one range camera. The RGB-camera is a color sensor capable of measuring color in

its field of view. The range camera is an infrared (IR) depth sensor that collects signals from the IR emitter on the Kinect™ as well. A schematic of the Kinect™ sensor can be found in Figure 8.

Figure 8: Kinect™ Sensor Schematic [20].

Although Microsoft has not released any official hardware specifications for the Kinect™, the OpenKinect community [18] has calculated the following specifications. The Kinect™ has a field of view of 57º horizontal, 45º vertical, and 70º diagonal. It has a spatial x/y resolution at 2 meters distance from the sensor of 3 mm and an operation range of 0.8 meters to 3.5 meters.

2.3 Robot Operating System

The software found onboard each TurtleBot™ was the Robot Operating System (ROS). ROS is a set of software libraries and tools that help build robot applications. From drivers to state-of-the-art algorithms, ROS is a powerful developer tools that is open source [21]. ROS is the base by which the TurtleBot™ system operates and communicates. The ROS framework allowed each experiment to be constructed using Python coding language.

2.4 Wireless Sensor Network

In order to communicate with multiple robots at the same time, a Wireless Sensor Network (WSN) was implemented into this project. The WSN used in this research is comprised of an isolated Wi-Fi network used inside the laboratory. Through this network, a control computer operating in Linux, called the base station, uses a secure shell to securely login remotely to each robot computer located on each TurtleBot™. By remotely logging in, the base station is able to send and receive operating information of each robot. Remote login was used for most experiments performed in this thesis.

2.5 Research Approach

The research approach that was conducted for the purposes of this research was an experimental methodology. The reason this approach was used for this research was because experimental data was needed to understand the capabilities and limitations that a TurtleBot™ had in order to recreate a simple SSA problem in a 2D environment. By understanding the uncertainty of the robot system, a maneuvering target could be sensed and tracked, predicting its prior and posterior state using an estimation algorithm.

2.6 Data Collection Method and Tools

For the purposes of this research, distances from the sensor to an object were calculated using a metric tape measure. The data collected from the tape measure was considered for this research to be the true distance away from the sensor to an object. Distances were also collected using the Kinect™. These distances were considered, for this research, to be the measured distances from the sensor to an object.

Data for this research was collected both by computer code and by hand. When collecting measured distances, a computer code was created to sense distance using the Kinect™, which

stored all the information into a text file. When collecting true distances, measurements were collected by hand in a notebook, and then recorded onto MatLab®.

In order to assure all runs in an experiment started at the same location and orientation, the laboratory floor tiles were used as a grid system to assure consistency. Each floor tile had dimensions of 12 in x 12 in. The floor tiles allowed for proper alignment when starting a run, and allowed for accurate metric tape measurement when a run would end.

2.7 Experimental Test Setup

For this research project, five different experiments were performed in understanding the basics of a SSA problem. Each experiment focused on a different objective, all leading to further understanding of characterizing uncertainty with the sensors onboard each TurtleBot™. The following describes each experimental test setup used in this project.

- *Test 1: Range Calibration*

  The range calibration test involved setting up one TurtleBot™ at a fixed location and orientation on the laboratory floor. A box was then placed at a specific distance away from the Kinect™ sensor onboard the TurtleBot™. The distance between the box and the sensor was then increased incrementally. At each distance, the TurtleBot™ would collect both range and bearing measurements. Since the box was placed at the center of the field of view of the robot, bearing measurements collected were near zero and were ignored for this test. This allowed the correlation coefficients of range measurements to be analyzed as a function of distance. Figure 9 describes the experimental setup used for this test. The black solid lines in this figure represent the box used for calibration.

- *Test 2: Bearing Calibration*

The bearing calibration test involved setting up one TurtleBot™ at a fixed location and orientation on the laboratory floor. Another TurtleBot was then placed at a specific distance (see Y axis in Figure 10) away from the Kinect™ sensor onboard the stationary TurtleBot™. For each run, the distance between the sensor and the robot's path was increased incrementally. The sensing TurtleBot™ would measure both range and bearing of the moving object. For this test, the objective was to have the moving TurtleBot™ travel through the entire field of view of the fixed robot to analyze the bearing limitations and uncertainty found with the Kinect™ sensor. Figure 10 describes the experimental setup used for this test.

- *Test 3: Circular Orbit Calibration*

  The circular orbit calibration test involved setting up one TurtleBot™ at a fixed location and orientation on the laboratory floor. Another TurtleBot™ was then placed at a specific radial distance away from the center of the stationary TurtleBot™. The maneuvering robot was then programmed to travel in a circular path creating a circle path of radius 2.37 ft. The objective of this test was to understand the tracking capabilities that the TurtleBot™ had at following an object moving in a circular path. Figure 11 describes the experimental setup used for this test.

- *Test 4: Orbit Tracking using Multiple Sensing TurtleBots™*

  The orbit tracking test using multiple sensing TurtleBots™ involved setting up three stationary TurtleBots™ on the edge of a 1 ft. radius circle. There was then a fourth TurtleBot™ placed on an outer circle of radius 2.37 ft. The inner circle in this test represented the surface of the Earth and the outer circle represented the path of an orbiting object. The maneuvering robot on the outer circle was programmed to travel around in a

circular path, going through the field of view of each TurlteBot™ progressively. The objective of this test was to setup a simple SSA problem where multiple sensors would collect measurements of an orbiting object. Figure 12 describes the experimental setup for used this test.

- *Test 5: Orbit Characterization using Extended Kalman Filter*

  The orbit characterization test using an Extended Kalman Filter involved a sensing TurtleBot™ at a fixed location and orientation while another TurtleBot™ traveled about a quarter circle path around the center of the fixed TurtleBot™. The radial distance of the moving robot's path was approximately 5 ft. The objective of this test was to assess the use of a mathematical estimation filter that would output the measured position, calculated state, and calculated predicted positon of the moving TurtleBot™. Figure 13 describes the experimental setup used for this test.

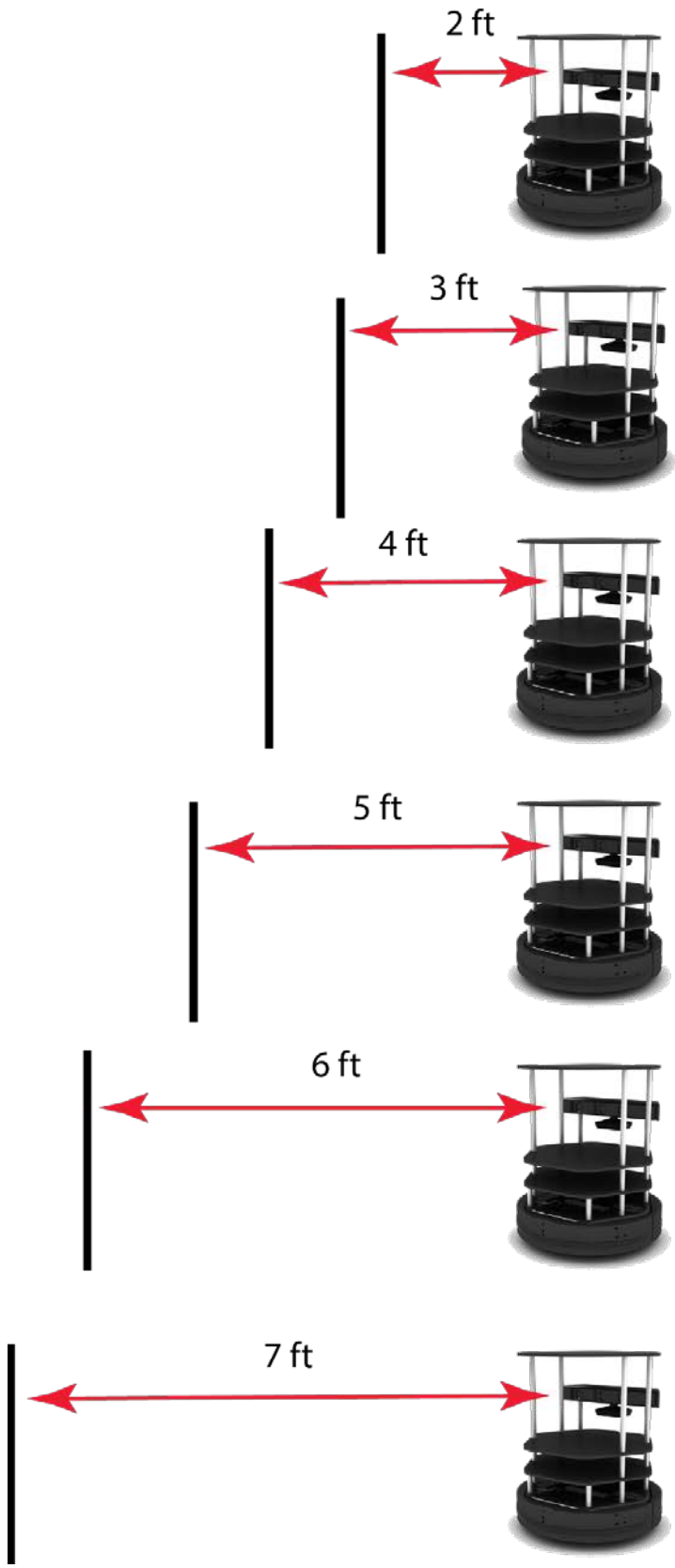

Figure 9: TurtleBot™ Range Calibration Test Diagram

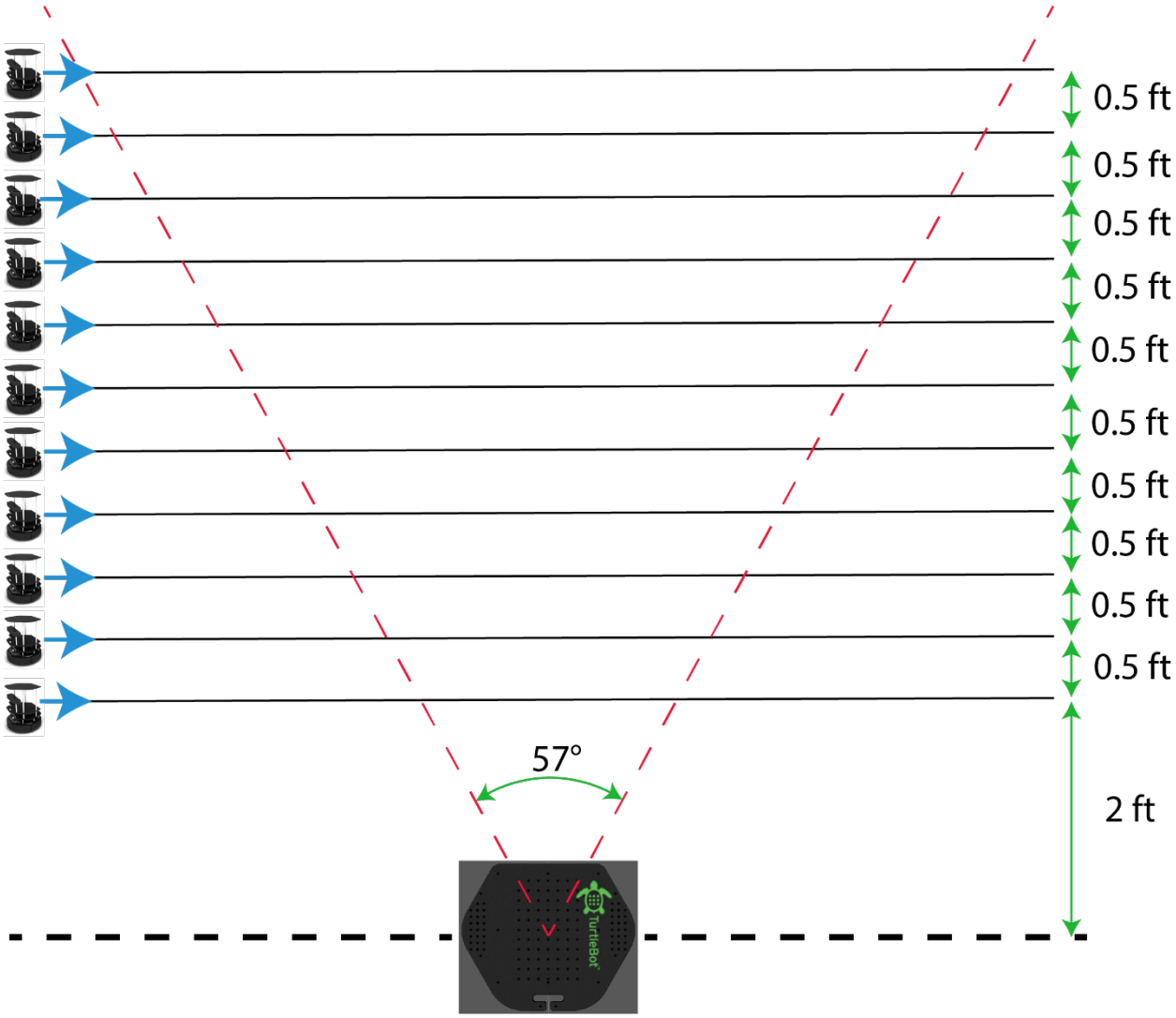

Figure 10: TurtleBot™ Bearing Calibration Test Diagram

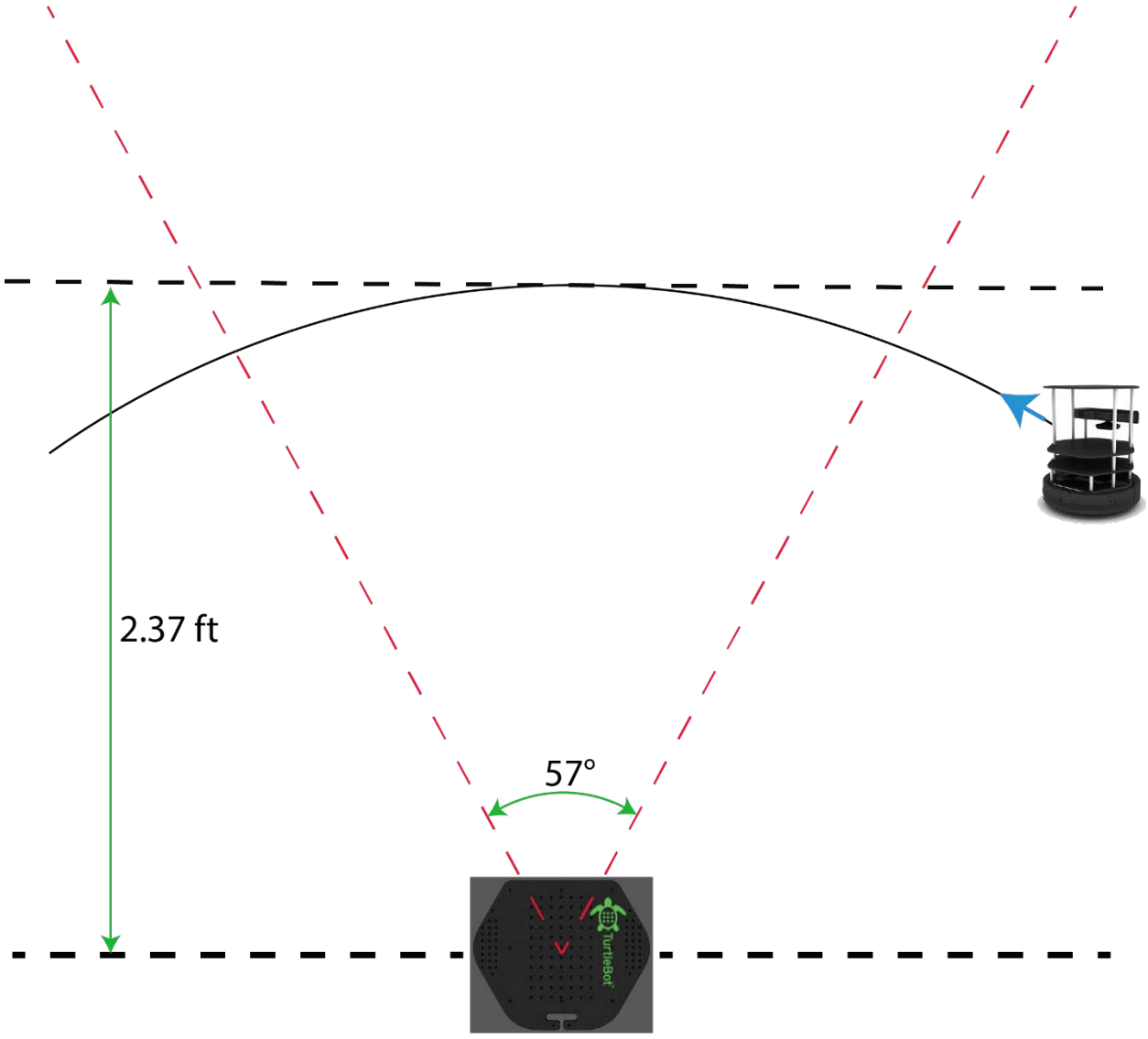

Figure 11: TurtleBot™ Circular Orbit Calibration Test Diagram

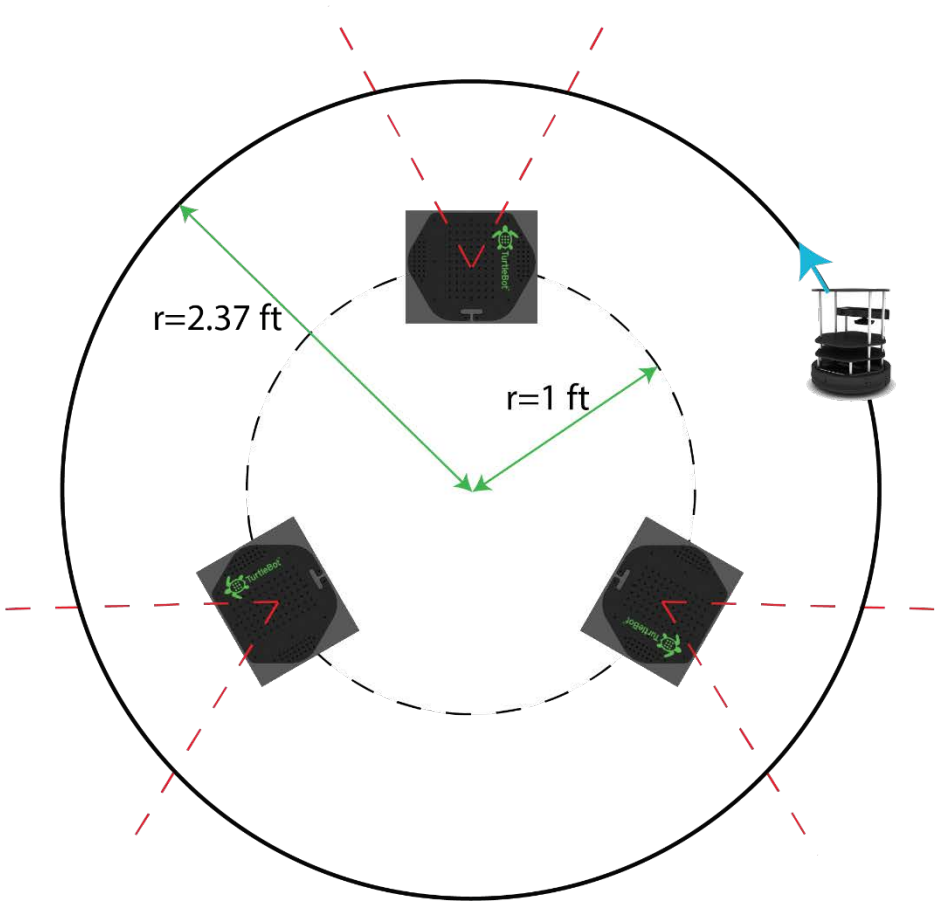

Figure 12: Orbit Tracking using multiple Sensors Diagram

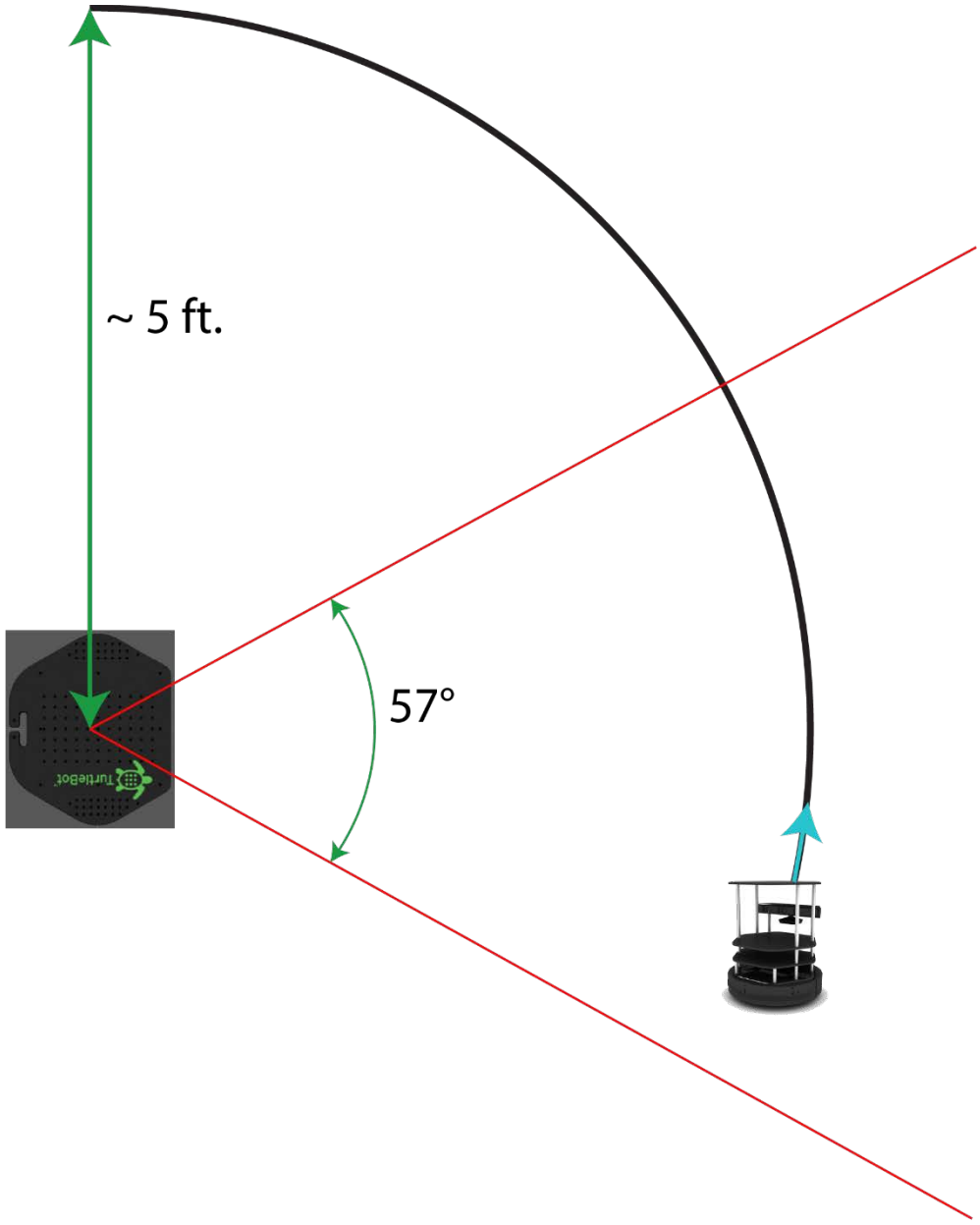

Figure 13: Orbit Characterization Test Diagram

2.8 Research Process

All research experiments were performed at the Laboratory for Autonomy in Data Driven and Complex Systems housed at the Aerospace Research Center of The Ohio State University, a facility that was built around robotic systems testing. A total of four TurtleBots™ were used throughout this research, all operating on software built on the ROS and communicated over a

wireless network to a workstation computer called the base station. The laboratory provided space for all experiments, robots, sensors, and the workstation computer. Experiments were conducted over a span of three calendar months.

2.9 Data Analysis

All data analysis in this research was performed using MatLab®. Data that was collected by the Kinect™ was saved onto the TurtleBot™ robot computer as a text file. This file was then placed on a USB storage drive and transferred over to a computer with MatLab®. The text file was then read on MatLab® using the *importdata* function. Data that was collected by hand and saved in a notebook was entered manually into MatLab®. All the results presented in this thesis were analyzed and created using MatLab®.

2.10 Extended Kalman Filter

An Extended Kalman Filter (EKF), or estimation filter, was used in this research to estimate the prior and posterior state position of a maneuvering target. This filter used both state and measurement predictions to calculate a posterior data point and then update the parameters found in the estimation filter. Using the first measurement of a sensor, a state propagator was created and used in the predicted state of the moving object. The dynamics matrix used in this filter was a circle because the motion of the robot was anticipated on be circular. Then, the state prediction was used for a measurement prediction. The Jacobian of the observation model had to be taken to predict a measurement. In the EKF, it was assumed that the dynamic noise covariance was zero mean and equal to the covariance matrix Q. Matrix Q was made up of all zero values. It was assumed that the measurement noise covariance was zero mean and equal to the covariance matrix R, but matrix R was created from the range calibration test data. The Kalman gain of this filter was chosen by default. Once all of these parameters were collected, the filter outputted a posterior state vector and covariance value. These two parameters were then used to update the

filter and the process repeated itself. For this research, the Kalman filter was only used for 5 time steps of 1 second lengths. For mathematical details of this filter, please see Appendix C.

2.11 Research Limitations

As it is for every research project, this research had the following limitations:

- *Time limitation* – The research was conducted in only two academic semesters. This project took time to understand and analyze the problem trying to be solved. Fitting this entire study into a short period of time was a big challenge to overcome.
- *TurtleBot™ limitation* – The research only had four TurtleBots™ to work with. This became limiting when technical issues were encountered with multiple robots and experimental runs could not be performed.

CHAPTER 3 – RESULTS

3.1 Overview

A TurtleBot™ could detect an object moving in front of its field of view, and also track the object using IR sensor measurements. Data was collected from this IR sensor and was then plotted to understand what the TurtleBot™ had seen in its field of view. Calibration of sensors was an important portion of running experiments to understand the capabilities that these sensors had when measuring in their field of view and its uncertainty. Results did show that as distance between the sensor and the target object increased, so did the noise in the data. Calibration results of both range and bearing are described in this chapter.

Error within the data that was collected from each experiment performed was a critical portion of the results described in this chapter. For each experiment that was performed in the laboratory, error could be found in the data in multiple ways. One area that was interesting to analyze was the motors that were found on the Turtlebot™. Motors would produce inconsistences

that could be visually seen when the data collected was plotted. This meant that the data that was collected had to be carefully examined to understand the error that was being produced from each run.

3.2 Range Calibration Test

The first test that was conducted with the TurtleBots™ was a range calibration test. This test allowed the TurtleBot™ to measure the X distance of an object at a stationary point in the field of view of the robot (see Figure 9). Six different distances (2 ft., 3 ft., 4 ft., 5 ft., 6 ft., 7ft) were used for the first range calibration test. At each distance, the TurtleBot™ collected 500 distance samples at 5 Hertz. The mean of the data points at each distance was calculated and then plotted against the correlation coefficient of each true distance. The correlation coefficient was calculated between the range measurement and the bearing measurement from the Kinect™. A linear trend line for association was added between the correlation coefficients as a function of mean X distance. Results from the first calibration test can be seen in Figure 14.

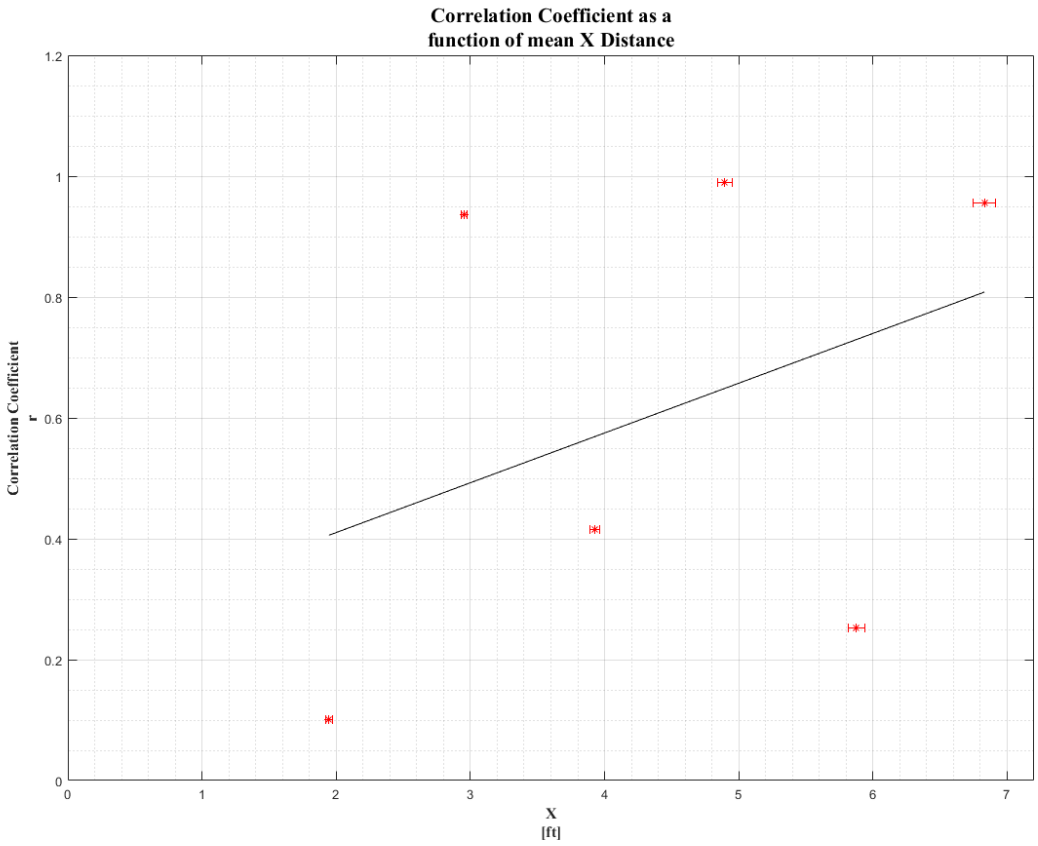

Figure 14: TurtleBot™ Range Calibration Test 1

The second range calibration test that was performed with the same TurtleBot™ used eleven different distances (2 ft., 2.5 ft., 3 ft., 3.5 ft., 4 ft., 4.5 ft., 5 ft., 5.5 ft., 6 ft., 6.5 ft., 7 ft.) away from the robot sensor. The reason more distances were included in the second range calibration test was to minimize the amount of error taken from these range observations and further understand if there is a relationship between correlation coefficient and measured mean X distances. Results from the second range calibration test can be seen in Figure 15.

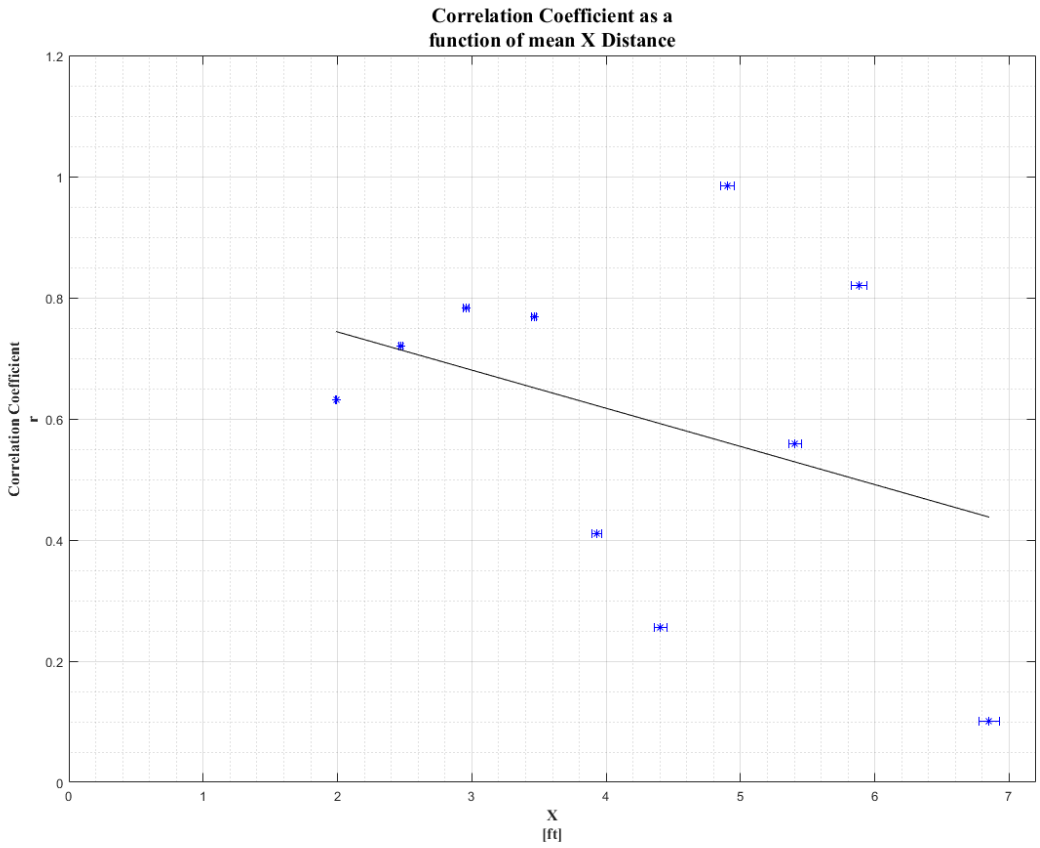

Figure 15: TurtleBot™ Range Calibration Test 2

As seen in Figure 15, at 6.5 ft. the TurtleBot™ sensor encountered an issue and data at that distance could not be analyzed and plotted. Examining both Figure 14 and 15 shows that there was no clear association found in correlation coefficient as a function of distance. If a clear trend was present, it could have been assumed that the correlation coefficient between range and bearing did change with distance. Error bars in both figures also describe that accuracy in the distance measurement decreased as the X distance increased. The overall error found between true distances and measured distances in the range calibration tests was about 6%.

3.3 Bearing Calibration Test

Due to equipment malfunctions, only one bearing calibration test could be completed. The bearing test consisted of having one TurtleBot™ serve as the stationary sensor (X=0 ft., Y=-2 ft.)

while a target crossed its field of view from left to right in a linear pattern (see Figure 10). The target, in this case another TurtleBot™, would be programmed to run across the entire field of view of the sensor at a speed of 0.05 meters per second. This was a measurable velocity programmed only in the +X direction. This test served to experimentally test the capability that the sensing TurtleBot™ had with the Kinect™. Since there was no official specification document found for the Kinect™, this test helped note the available bearing sensing capability that the stationary TurtleBot™ had. This experiment consisted of eleven runs all at different distances from the Y=-2 ft. line (2 ft., 2.5 ft., 3 ft., 3.5 ft., 4 ft., 4.5 ft., 5 ft., 5.5 ft., 6ft., 6.5 ft., 7 ft.). Results from this test can be found in Figure 16.

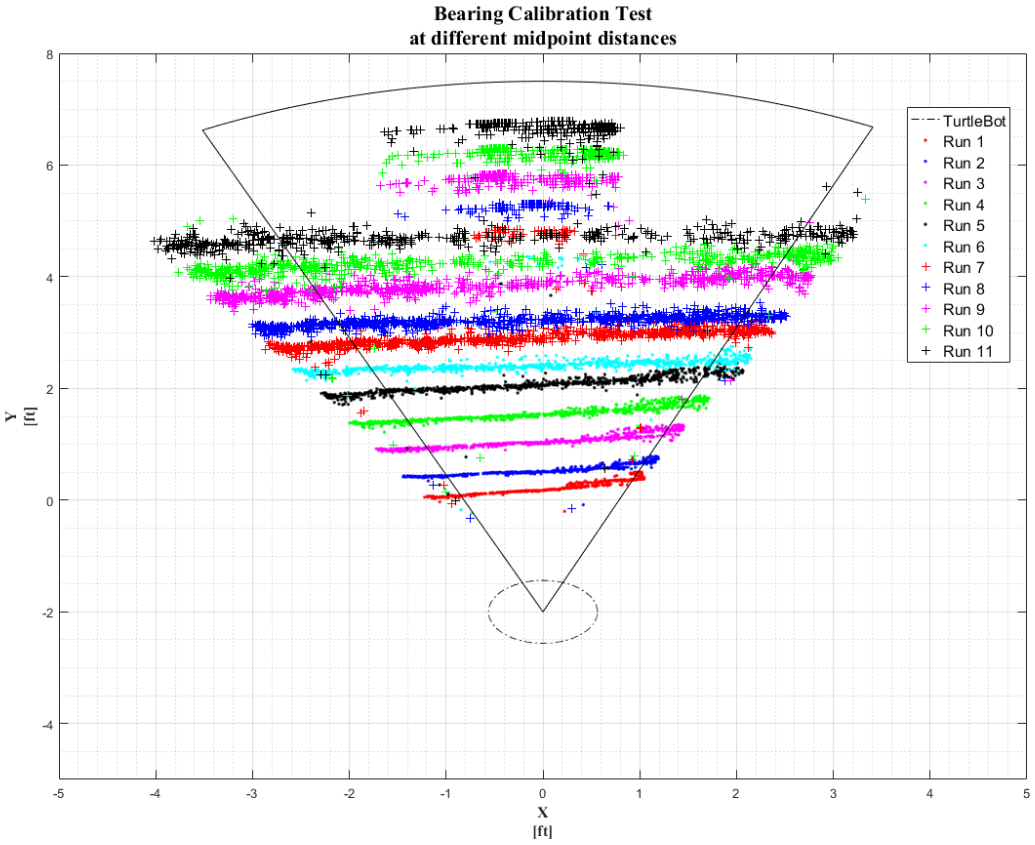

Figure 16: TurtleBot™ Bearing Calibration Test

As seen in Figure 16, as the distance increased between the sensing TurtleBot™ and the maneuvering target TurtleBot™, noise did increase from the sensor. Outlined in the solid black lines in Figure 15 is the mean of the maximum and minimum bearing measurements taken from all the runs. The mean field of view angle was calculated to be 57.29º. The outlined bearing range did show positive information about the capability that the TurtleBot™ has using the Kinect™ as a mean for range and bearing measurements. The overall error found between true distances and measured distances in the bearing calibration test was 7.5%. Comparing results from both the range and bearing calibration tests, results prove to be highly accurate.

Another important aspect noted from Figure 16 is that there was a slight deviation in all the linear runs from the target TurtleBot™. This deviation was analyzed through a histogram plot that showed how from the linear path was the target TurtleBot™ moving (see Figure 17). Three different distances (2 ft., 3 ft., 4 ft.) were chosen to be analyzed to further understand the error found in the data.

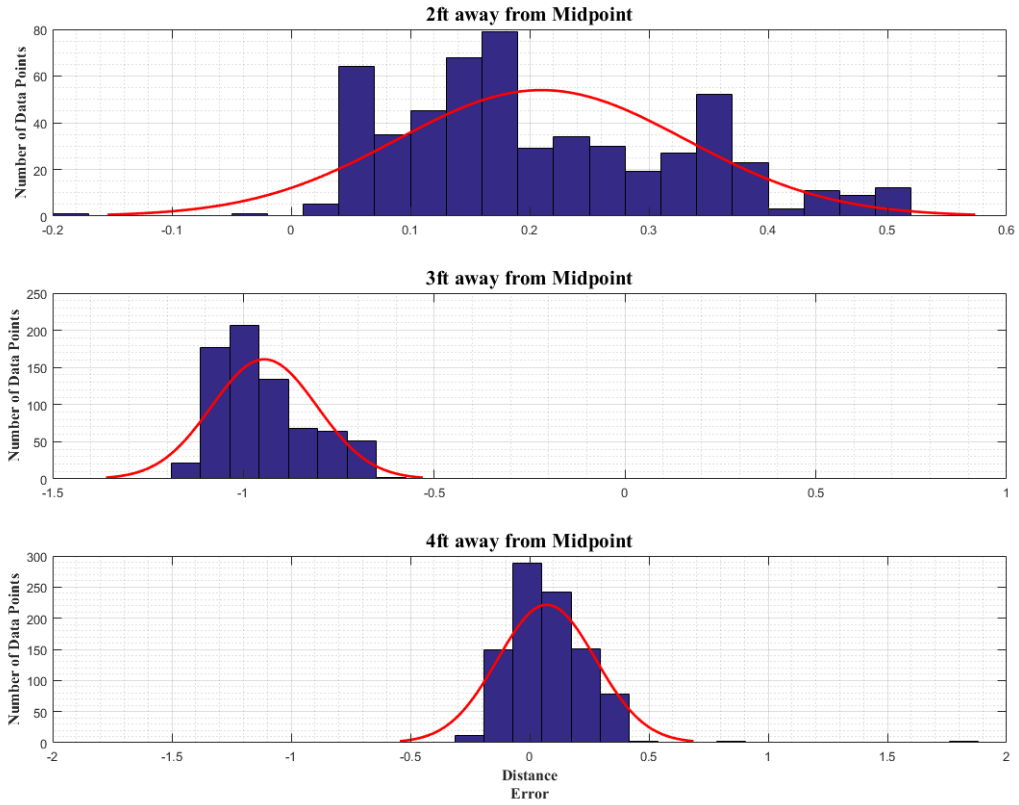

Figure 17: TurtleBot™ Bearing Calibration Test Deviations

No specific deviation was found from Figure 17, but it is evident that no test was perfect. Deviations were not dependent on distance. At a 2 ft. midpoint distance, the moving target overshot the targeted linear path. At a 3ft. midpoint distance, the moving target undershot the targeted linear path. At a 4 ft. midpoint distance, the moving target was the closest to the true distance. The deviation presented in this experiment offered information that the motors found on the Turtlebot™ had some prescribed error to keep in mind for future measurements.

3.4 Circular Orbit Calibration Test

To further understand the experimental limitations that a TurtleBot™ would have in correspondence with orbital debris research, a circular orbit calibration test was performed using a sensing TurtleBot™ and a moving target TurtleBot™. This experiment was critical because this was the first step in executing and analyzing a simple SSA problem. This test recreated a sensor

site, the sensing TurtleBot™, on the surface of Earth while an object, a moving TurtleBot™, orbited the Earth. The sensing robot had to measure both range and bearing of the moving object as it crossed its field of view in a circular path (see Figure11). The results of this test can be seen in Figure 18.

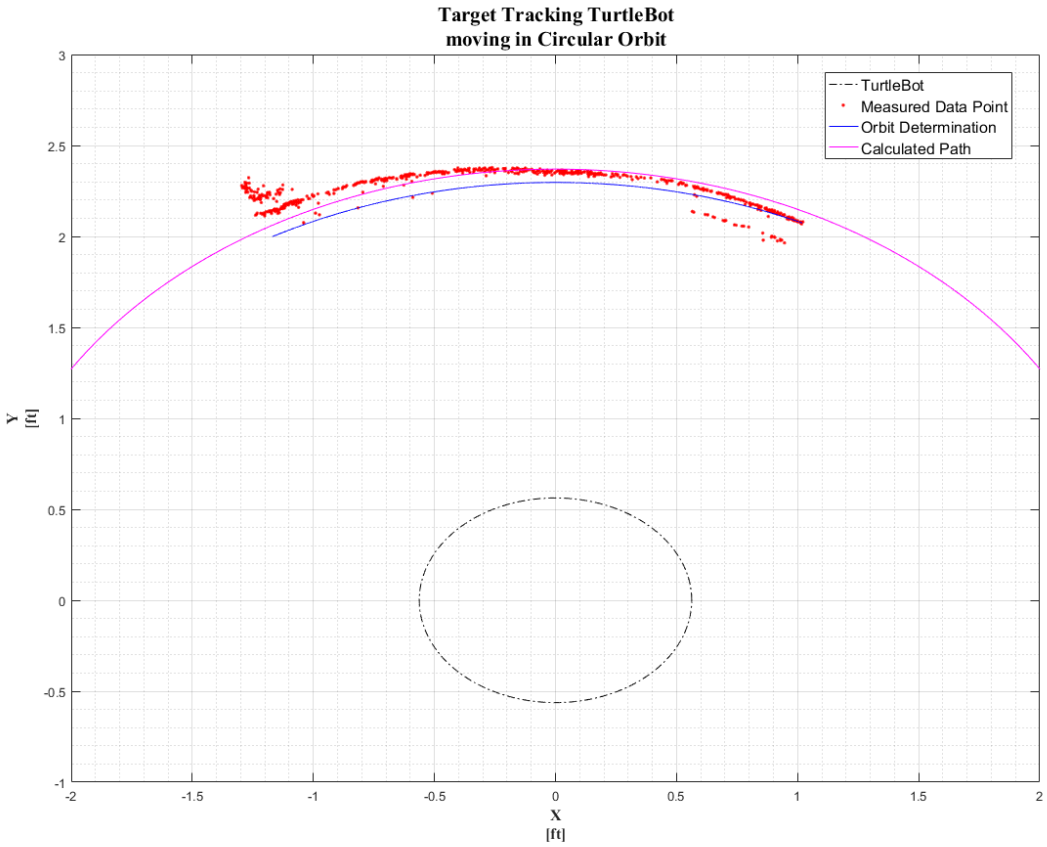

Figure 18: TurtleBot™ Circular Orbit Calibration Test

For this test, the moving target TurtleBot™ had a linear speed of 0.05 meters per second and an angular speed of 0.05 meters per second. Using this information, the calculated orbit radius for this experiment was 2.37 ft. from the center of the sensing TurtleBot™. The calculated orbit of the moving target can been see in the solid magenta line presented in Figure 18. For this experiment, an orbit determination code (found in Appendix A) was created to analyze the data collected to an orbital mechanics perspective. This included being able to map the trajectory of an

orbit and present orbital parameters. These parameters included eccentricity, semimajor axis, and angular momentum. The orbital trajectory (see Appendix B) plotted from the data collected from this experiment can be seen in the solid blue line in Figure 18. The error found between the calculated target path and the orbital trajectory from measured values was 2.85%. This means that the orbit determination code was highly accurate for a scaled SSA experiment.

Another important aspect to note from the circular orbit calibration test is that there was a small deviation noted from the measured values to the calculated target path. To further investigate this deviation, the true circular path was compared to the measured path by means of a histogram plot. The deviation found from measurements were analyzed for three consecutive orbits around the sensing TurtleBot™. This data is represented in Figure 19.

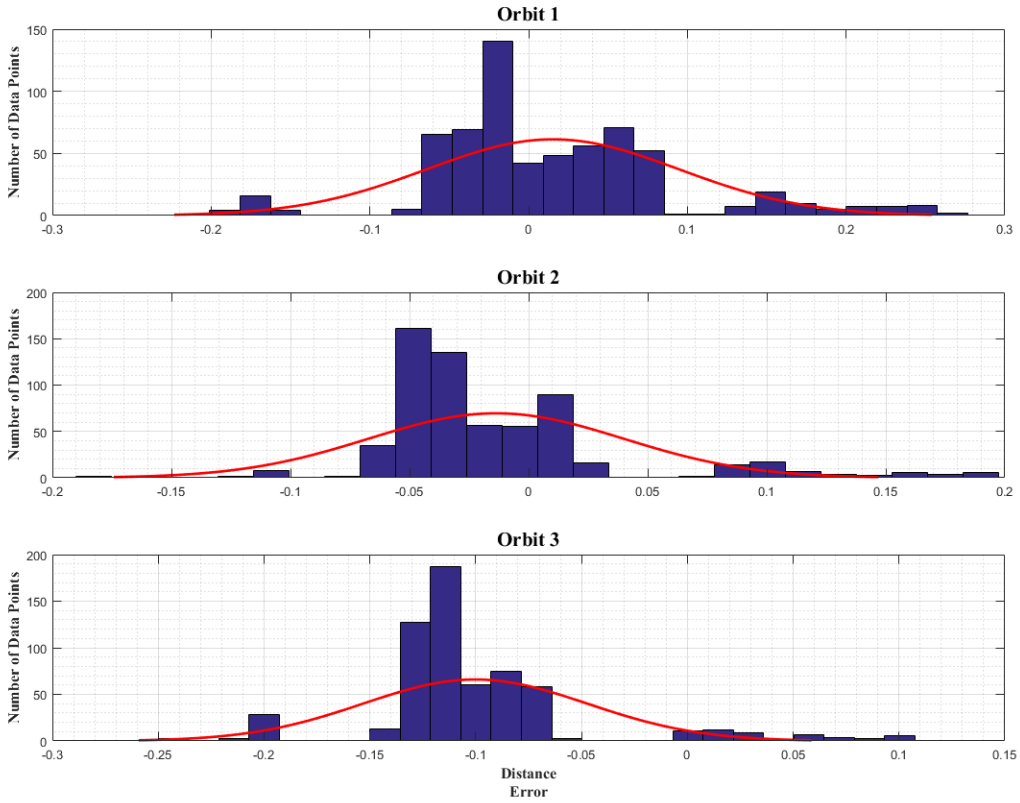

Figure 19: TurtleBot™ Circular Orbit Calibration Test Deviations

Figure 19 represents the deviation found between the calculated target path and the measured target path. It is evident that as the moving TurtleBot™ orbited the sensing TurtleBot™, a small deviation began to form causing the orbit radius of the circling robot to decrease slightly. Although this is a small deviation, this error must be considered when tracking a target and when programming a robot to orbit a sensor site. It is predicted that this deviation occurred due to error found in both linear and angular velocities created by the robot motors. By considering these small errors, the SSA problem can be formulated to be more realistic.

3.5 Basic Space Situational Awareness Problem Test

To further enhance the realism found in this research, one experiment that was created to replicate the current SSN and orbital debris tracking was a multiple sensor test tracking one target moving around in a circular path. This was done using three TurtleBots™ placed at the edge of a 1 ft. circle imitating the surface of the Earth. These TurtleBots™ were spaced equally at 120° and represented sensor sites currently found around the world. A fourth Turtlebot™ was used to denote orbital debris traveling around the Earth (see Figure 12). This maneuvering robot was programmed to travel at a linear speed of 0.05 meters per second and had an angular speed of 0.05 meters per second. This information calculated the orbit of the robot to be 2.37 ft. from the center of the Earth representation. The measured data from this test can be found in Figure 20.

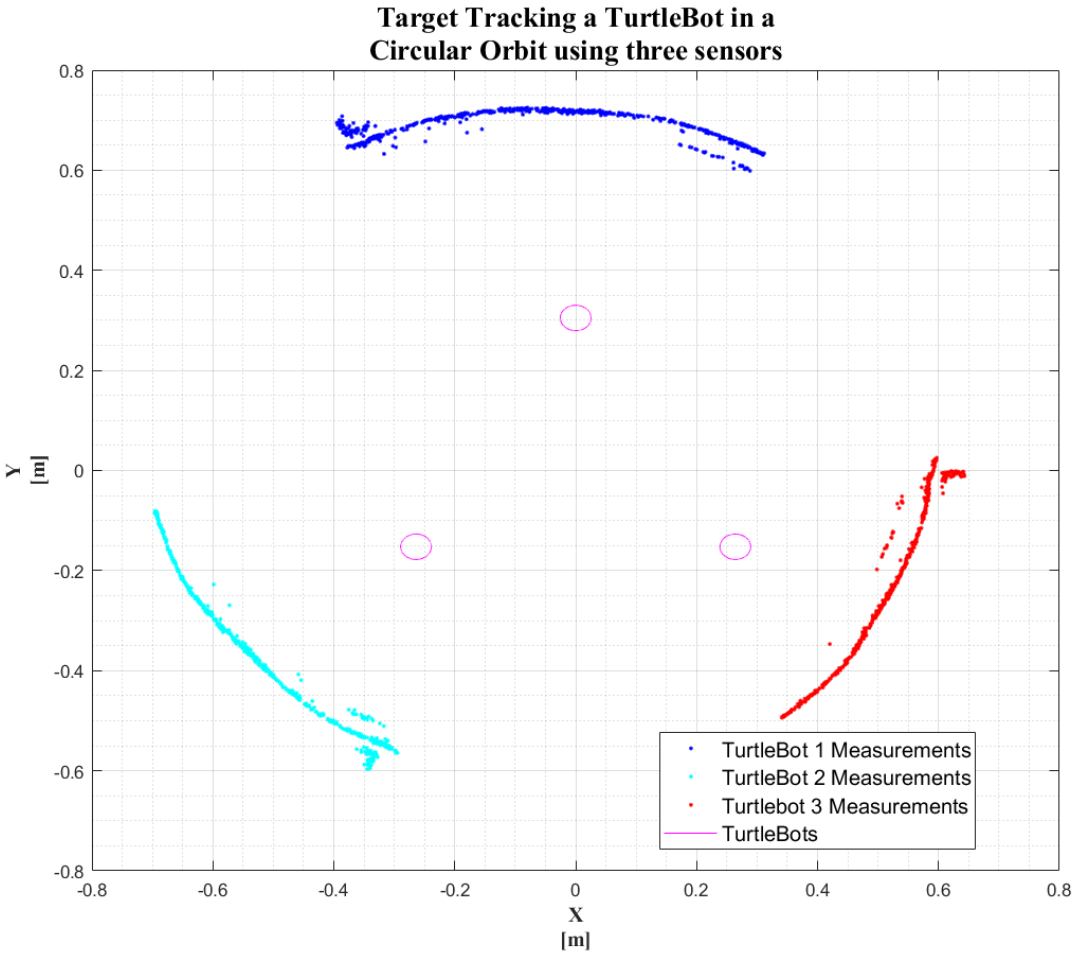

Figure 20: Space Situational Awareness Problem using three TurtleBots™

Figure 20 represents a simple SSA problem where a single orbiting object is being tracked by multiple sensors. This test was used as a proof of concept that multiple sensors could be used to track, and in the future, predict where an object will be as it orbits the Earth. Measured data from each TurtleBot™ is represented in this figure using different colors. Something to note about this test was that at the edges of the field of the view of each robot, more noise in the data in presented than in the middle portion of the field of view. This could serve of help in future experiments when the reliability of a certain sensor will be calculated to track a moving target. More dependable information could be used only when the object is in the middle portion of a

sensor's field of view. This would allow estimation filters being used to predict the position of a target more accurately.

3.6 Orbit Characterization using Extended Kalman Filter Test

To better understand orbital debris tracking methods, an Extended Kalman Filter (EKF) was implemented into one experiment to observe tracking and predicting capabilities using measured values. This test involved a stationary sensing TurtleBot™ who would track another TurtleBot™ traveling in a circular path (see Figure 13). The moving robot had a linear speed of 0.5 meters per second and an angular speed of 0.37°/sec. These parameters produced a traveling circle radius around the stationary robot of about 5 ft. The sensing TurtleBot™ recorded data for this test at a frequency of 50 Hertz for 5 seconds. This meant six data points were collected for each measured run. A total of nine runs were recorded for this experiment. Data collected can be found in Figure 21.

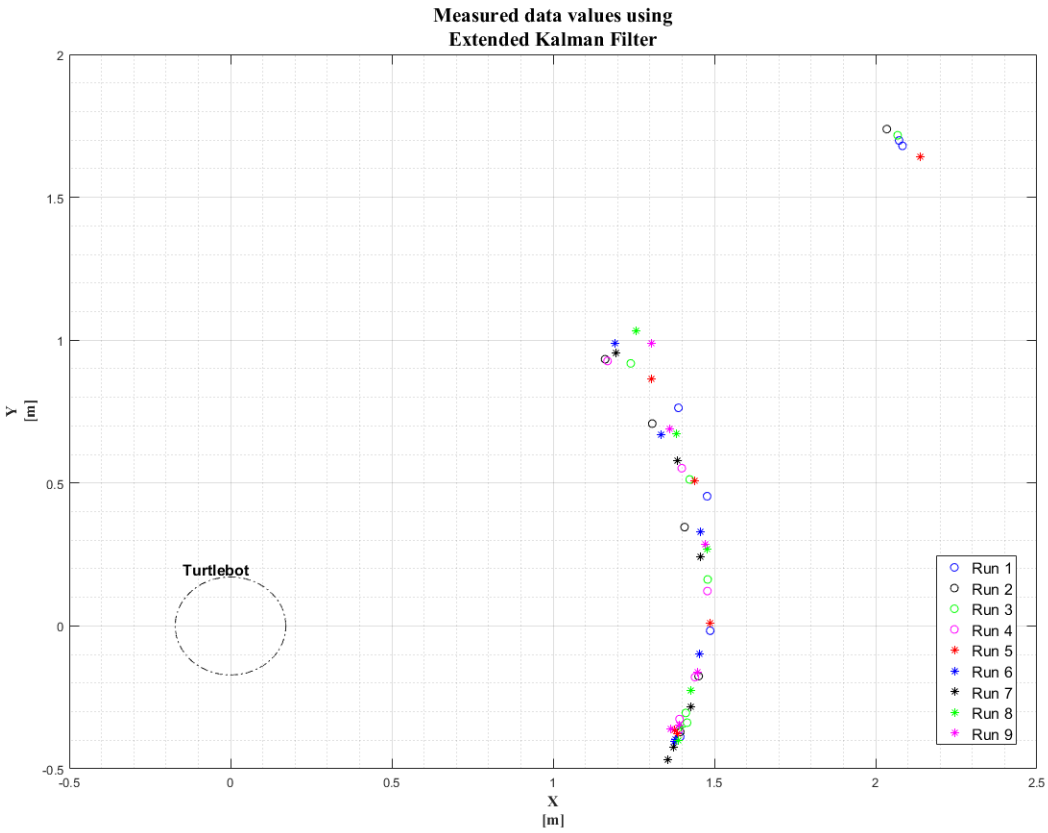

Figure 21: Measured Position using EKF

Although all measured position data points were taken at the same frequency, one issue with these results is that not all runs were caught to start at the same starting point. This issue came about because the starting method of the sensor capturing and the movement of the orbiting robot was inefficient. The sensing TurtleBot™ would be programmed to start collecting sensor data and afterwards the moving TurtleBot™ would be programmed to start traveling in a circular path. This became a challenge when the WSN would cause the command sent to the robot to delay because the signal was poor. This matter is what caused this data set to be inconsistent and would need to be repeated in the future for a more accurate analysis.

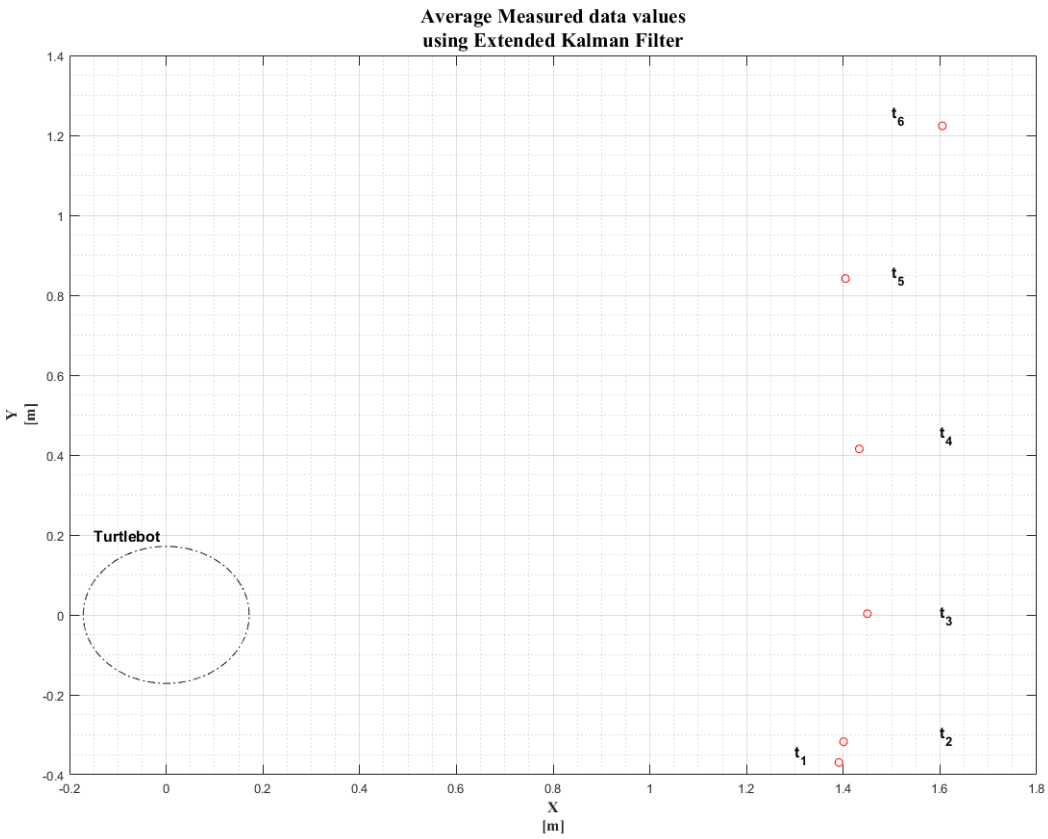

Figure 22: Mean of Measured Position using EKF

By taking the mean of measured position values at each time step, a clearer plot is presented in Figure 22. This data plot did consider an extra data point at the beginning of data collection from the sensor. Since the starting point of the moving TurtleBot™ was different for some runs, the target object was out of the field of view for the last time step which did factor in for the big position deviation on time step 6 in Figure 22. It is also noted from this plot that the EKF could capture the curvature of the path from the moving robot between time steps 1-5.

Besides being able to measure the position of the moving robot, the sensing TurtleBot™ was programmed to capture the state of the target moving through the EKF. The EKF used simple equation models (see Methodology) of a circular path to be able to take the measured values and

transform them to state data points. The data collected for the state of the moving object can be found in Figure 23.

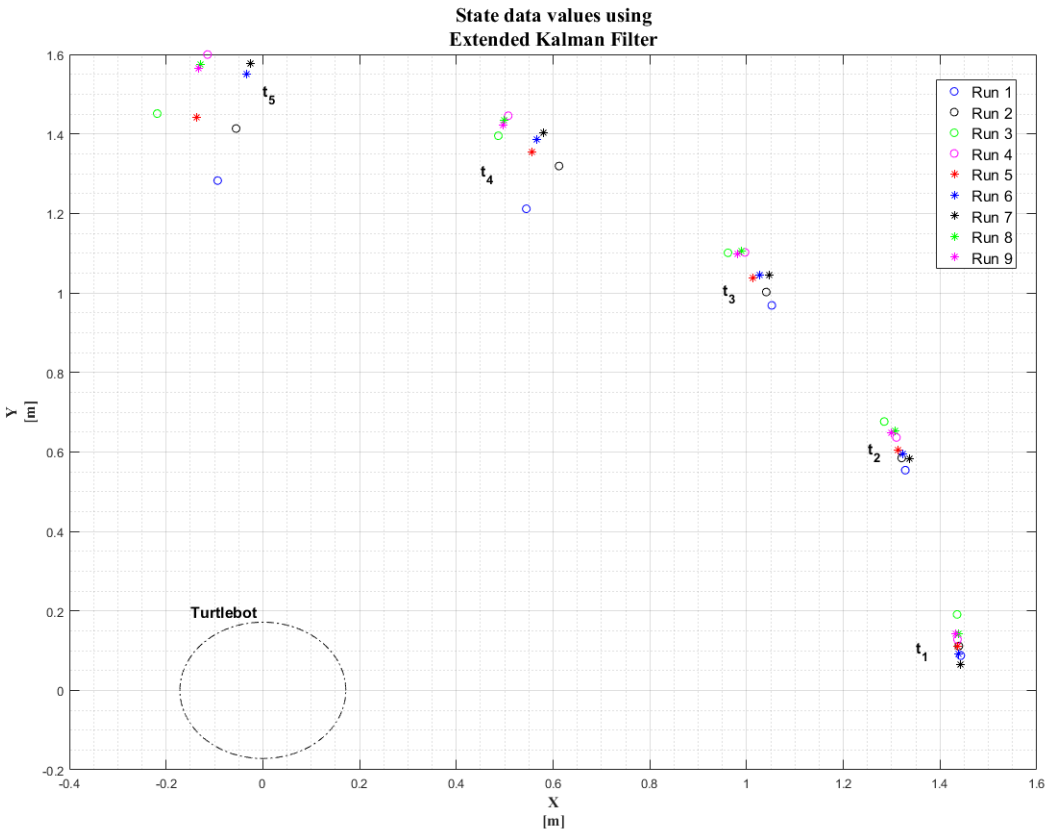

Figure 23: State of moving object using EKF

The state of the moving object was calculated by the EKF at five different time steps. Each time step is a one second interval. It is evident from this plot that the state equations that were used did represent a circular path of the moving target. By understanding the state of the moving target, the predicted position of the object could be found.

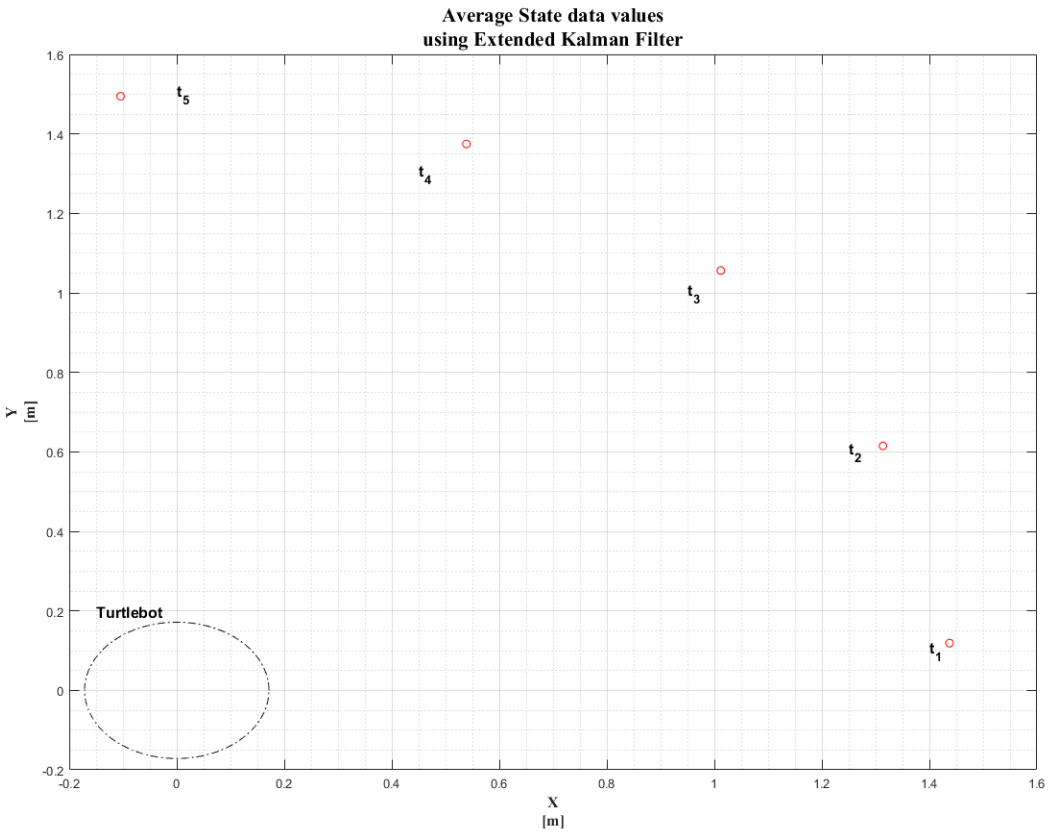

Figure 24: Mean State of moving object using EKF

The mean state of the object was taken at every time step and then used to present the motion of the target. The mean data points can be found in Figure 24. This plot visibly describes the motion of the moving object to be a quarter of a circle.

By calculating the state of the moving TurtleBot™ the EKF was able to then calculate the predicted position of the moving body. Taking the position observations collected from the Kinect™ and analyzing them with the calculated states values of the object, the EKF was then able to produce a predicted position for the target in the future. This basic prediction test allowed for an understanding of how accurate this estimation filter was at calculating the future position of a moving object. Data collected for this test can be found in Figure 25.

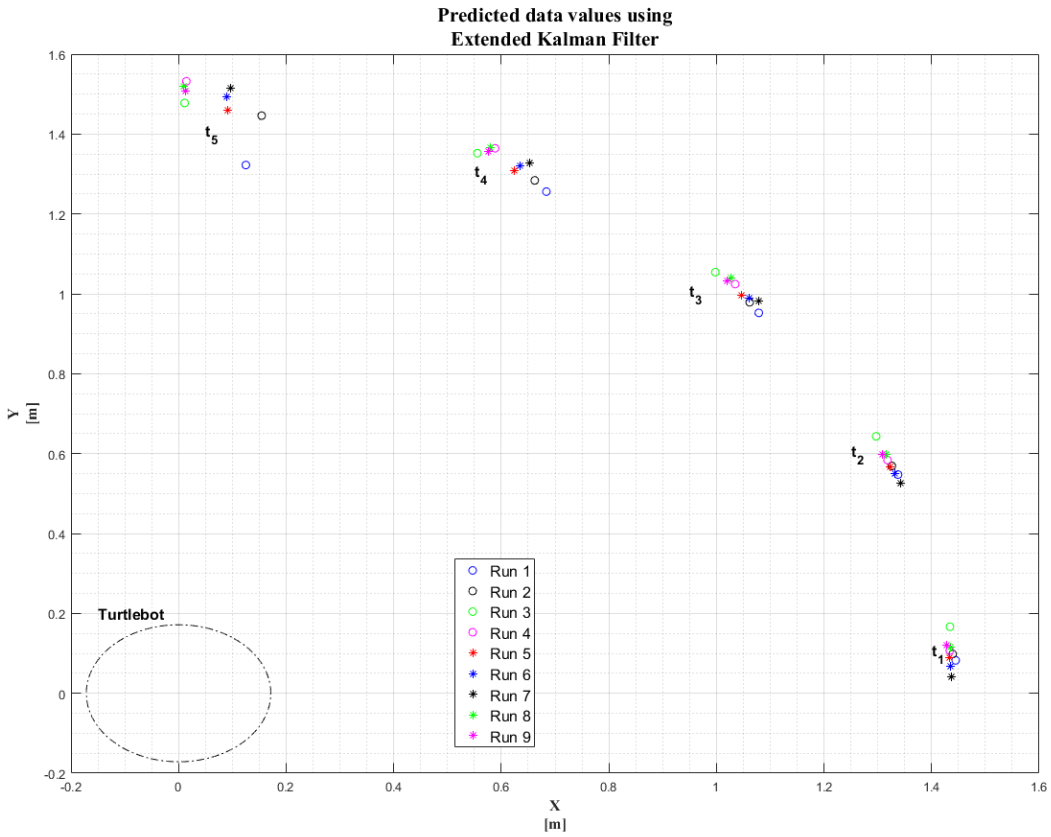

Figure 25: Prediction of moving object using EKF

Prediction data shows a clear trend that the motion of the moving object is a circular path. This is in combination with the calculated state data which means that the models being used in the EKF are accurate.

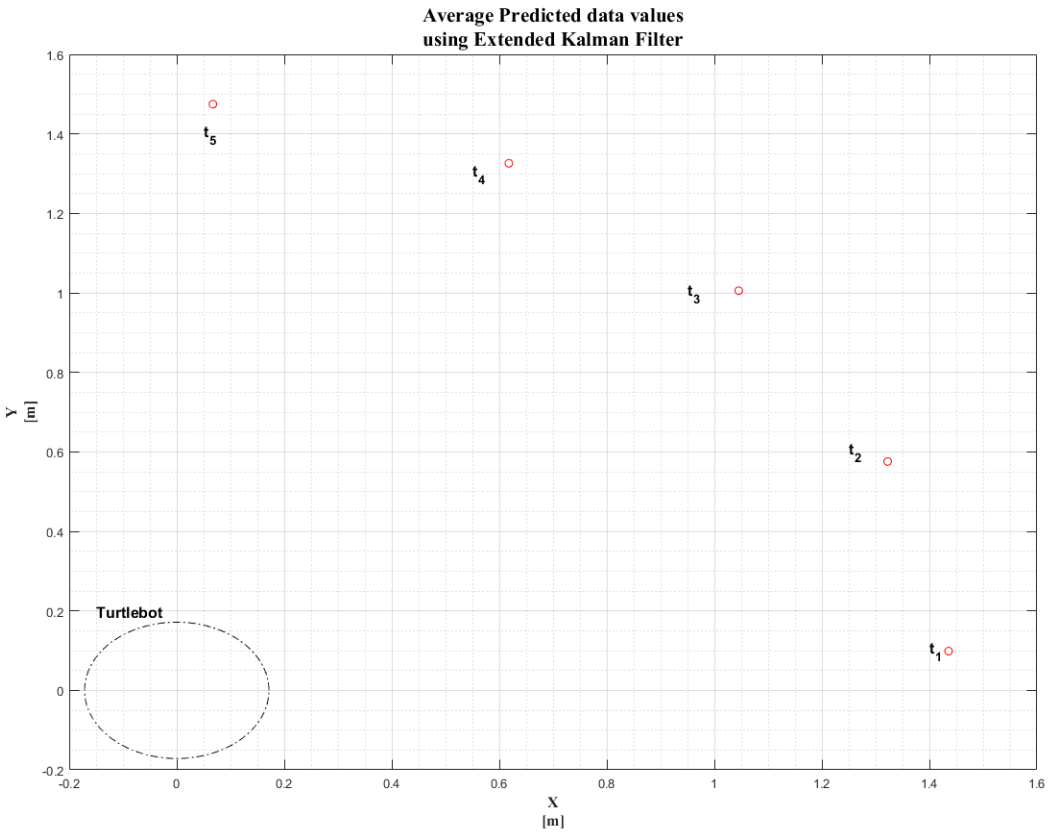

Figure 26: Mean Prediction of moving object using EKF

The mean prediction of the object was taken at every time step and then used to present the motion of the target. The mean data points can be found in Figure 26.

In further examining the relationship between measured, prior, and posterior values collected and calculated, it can be understood that the prior and posterior do fall within what was expected of a circular path of the target, but the measured values do have deviations. The measured values collected by the TurtleBot™ does initially show a curved path of motion, but then deviates towards the end of the route. Data points also show that the measured position of the moving body does not come align with what the prior and posterior position of the body is. This information is visible in Figure 27.

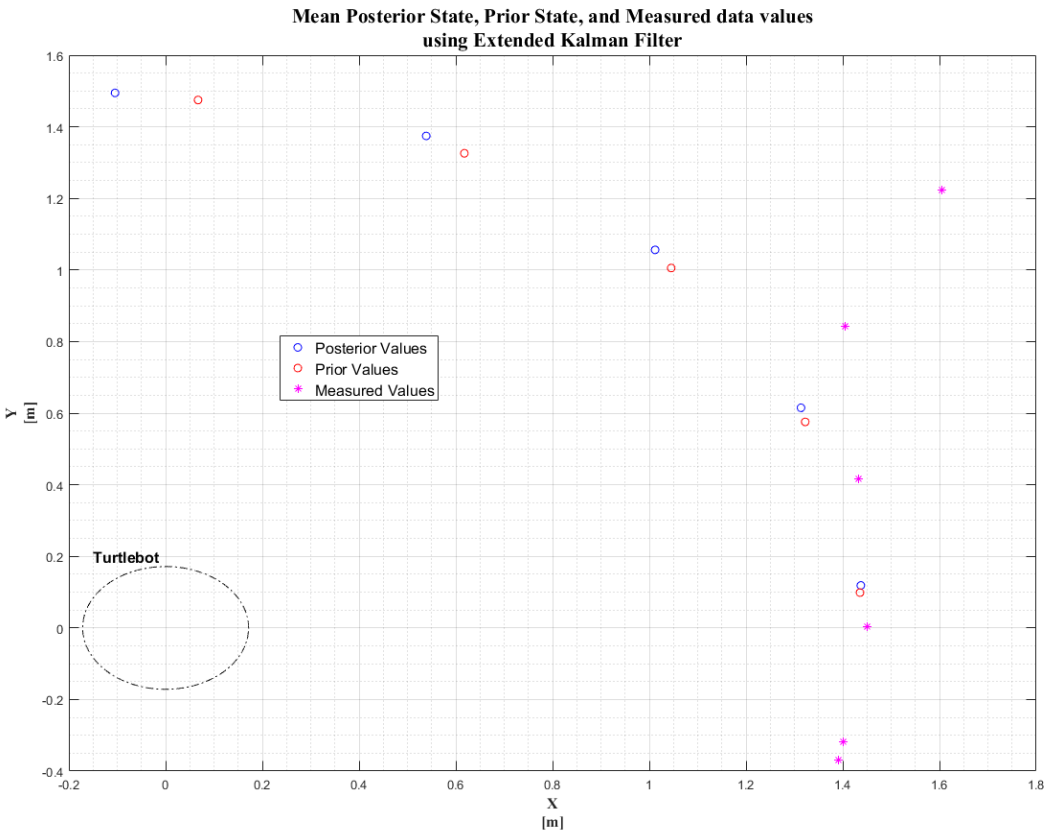

Figure 27: Mean Posterior, Prior, and Measured data of moving object using EKF

This plot describes the relationship between the measured, prior, and posterior values collected from the TurtleBot™. It is noted that prior and posterior values follow a similar trend of a circular path, but the measured values do not. As mentioned previously, error in measured values originated from inconsistent starting points for reach run. The timing between the start time of sensor and the start time of the moving object were not consistent. This inconsistency did affect the final results presented in this plot, as seen in the path deviation of the measured values. The last two measured values presented in Figure 27 also shows the need for covariance error ellipses. Error ellipses would help discard all unrelated information presented in this plot.

CHAPTER 4 – CONCLUSION

4.1 Contributions

To improve the current methods of orbital debris tracking, understanding the basics of what needs to occur to track an object is significant. This research explored the fundamentals of a SSA problem by replicated the SSN and an orbiting object in a laboratory environment. By setting up calibration tests, we were able to analyze how calculating the error found with the lab equipment could help improve an EKF that would not only measure the state of an object, but could predict where it would be in the future. The contributions that this research has provided so far will help set up more complicated SSA problems in the future. This research has furthered our experimental knowledge on the essentials of setting up a SSA problem in a laboratory setting. This research has provided insight on the kind of information that TurtleBots™ are capable of measuring and storing. This will become helpful in the future when data fusion techniques are explored to transfer this information from one robot to another.

4.2 Additional Applications

The experimental research that has been performed towards this project will allow for greater contributions outside of space applications. Target tracking has been implemented in disaster relief for search and rescue. Robots go into an environment and search for hazards for humans, allowing a rescue team to be more aware of their surroundings. These robots have sensors onboard that allow them to detect different features in an environment, such as distance, heat, vision, gas, and even blood [22]. The less uncertainty that is found in these sensors, the higher the probability of being able to save the life of a person in danger. The basic understanding of minimizing uncertainty in sensors will allow for more precise search and rescue technology in the future. This research explores the fundamentals to improve range and bearing measurements on a robot.

This experimental work could also have applications towards wildfire monitoring and predicting. Graduate students in this same laboratory are currently studying how wildfires grow and how drone technology could be implemented to track how long a wildfire will take to reach a certain location in an environment. The basics of characterizing uncertainty found in this research could help have an impact on that work.

4.3 Future Work

Although this research is not a new study, there are multiple areas of improvement that could be made to further understand orbital debris tracking. The first improvement for future work would be to run more experiments that were described in this thesis. More experiments would produce clearer similarities and differences between each experimental run. This would be significant because this would allow for further understanding of the limitations that are encountered using TurtleBots™ for a SSA problem. Multiple runs of a single experiment would also help account for outlier data that can be encountered with every run. By analyzing multiple runs, outliers from the data collected can be discarded producing more accurate results.

Another area of improvement found for this research would be to have each sensing TurtleBot™, described in Figure 12, wirelessly trade off measured data collected with one another to improve target tracking. Since the use of sensor sites in the SSN is expensive, the most efficient way to track orbital debris would be to turn on a sensor only when needed. This idea could be replicated with the SSA problem using three TurtleBots™. Each TurtleBot™ would be programmed to turn the following sensing TurtleBot™ on when an object disappears from its field of view. Once the orbiting object exits the field of view, that measured data collected could then be transferred over to the following TurtleBot™ wirelessly to allow it to detect and track the orbiting object. The previous sensor at that point would be turned off to save expenses.

Improvement to the determination of the gravitational constant G for orbital projections could also be investigated more. The current method of determining this variable works, but understanding how to replicate this constant at such a small-scale experiment is critical when this research is translated to real world space application.

Wireless connection improvements could all be made in this research. For some experiments, the wireless connection on which these robots operated in had a lag that allowed inconsistencies to occur in the data. This mean that sensors either collected data to earlier or too late, allowing a maneuvering object's position to be calculated incorrectly. Finding a better method to have one robot sense and another robot maneuver at the same time would be a big area of improvement for this project.

The implementation of covariance error ellipses in the data collected from the EKF would also help analyze the data. By applying error ellipses, unwanted data could be discarded leaving behind only the necessary data to track a moving object.

The last recommendation for future work on this research would be to account for process noise found in the EKF dynamics model. The dynamic model that was used in this research assumed a zero mean Q covariance matrix. By experimentally finding the Q matrix, a more accurate dynamics model could be created to plot the prior and posterior state of a moving object. This would results in a more accurate position estimation of a maneuvering target.

4.4 Summary

As humans continue to travel the space environment around Earth, orbital debris will continue to be a challenge for space exploration. Not only capable of being a hazard to humans, but to government assets as well, orbital debris needs to continue being observed and tracked. The information collected from observations needs to continue being stored in a database that will

allow multiple entities to access it in hopes to avoid high-speed collisions. If collisions do occur, the number of orbital debris increases, allowing the problem to increase that much more.

Since there are not clear steps to removing this orbital debris found the near Earth environment, the only steps that can be taken to discovering a solution is to find better methods of tracking this debris. Current methods work, but do not work efficiently. With a limited number of sensors placed around the Earth, and being expensive to operate, sensors must be operated in a more efficient way that will capture more data with less running time. To do this, it is hypothesized that by allowing the sensors to work on a single wireless network, it can be possible to have them communicate via this network and exchange data to track objects with more precision.

There are multiple actions that can be taken to minimize state uncertainty. Understanding the error found in the sensors is a fundamental start to diminishing uncertainty. As experimentally shown with this thesis, calibration tests do show the capabilities that the equipment being used has. Considering both measurement error in conjunction with estimation filters, the state uncertainty of a dynamically-moving target can be characterized to potentially decrease inaccuracy.

If positive actions are taken to help solve the problem of orbital debris, space exploration will be able to continue being pushed beyond its current limit. The lives of humans and space assets are on the line, but characterizing uncertainty in the technology that is currently available will help diminish accident probability.

APPENDICIES

Appendix A

Orbit Determination Code

```
function [ theta, mu, r, a, h, e ] = OrbitDetermination( earth, v_TT, avg_r, point1, point2,
point3 )

% Constants
r_E =       earth*0.3048;       % [km]
mu =        v_TT*(avg_r);
r =         avg_r;
theta =     linspace(0,2*pi,10000);

% Setting up x values
x1 =        point1(1)*cos(point1(2));
x2 =        point2(1)*cos(point2(2));
x3 =        point3(1)*cos(point3(2));

% Setting up y values
y1 =        point1(1)*sin(point1(2));
y2 =        point2(1)*sin(point2(2));
y3 =        point3(1)*sin(point3(2));

% Three R vectors
R_1 =       [x1 y1 0];
R_2 =       [x2 y2 0];
R_3 =       [x3 y3 0];

% Three R magnitudes
r_1 =       norm(R_1);
r_2 =       norm(R_2);
r_3 =       norm(R_3);

% Coefficients
C_12 =      cross(R_1,R_2);
C_23 =      cross(R_2,R_3);
C_31 =      cross(R_3,R_1);

% Unit Vectors
r_1_hat =   R_1/r_1;
r_2_hat =   R_2/r_2;
r_3_hat =   R_3/r_3;
C_23_hat =  C_23/norm(C_23);        %Z_p unit vector

% Checking to confirm 2D problem
is2Dprob =  dot(r_1_hat,C_23_hat);

% Finding N
N_vec =     r_1*C_23+r_2*C_31+r_3*C_12;
N_mag =     norm(N_vec);

% Find D
```

```
D_vec =     C_12+C_23+C_31;
D_mag =     norm(D_vec);

% Find S
S_vec =     R_1.*(r_2-r_3)+R_2.*(r_3-r_1)+R_3.*(r_1-r_2);
S_mag =     norm(S_vec);

% Combining informatoin to find velocity
V_2 =       ((mu/(N_mag*D_mag))^(1/2))*(cross(D_vec,r_2_hat)+S_vec);
v_2 =       norm(V_2);

% Semimajor Axis
a =         1/((2/r_2)-(v_2^2/mu));

% Angular Momentum
H =         cross(R_2,V_2);
h =         norm(H);

% Eccentricity
e_vec =     (cross(V_2,H)/mu)-r_2_hat;
e =         norm(e_vec);

end
```

Published with MATLAB® R2016b

Appendix B

Orbit Determination Equations: Gibbs Method 2D

Three Position Vectors

$$\vec{r}_1 = r_{1x}\hat{I} + r_{1y}\hat{J} \tag{1}$$

$$\vec{r}_2 = r_{2x}\hat{I} + r_{2y}\hat{J} \tag{2}$$

$$\vec{r}_3 = r_{3x}\hat{I} + r_{3y}\hat{J} \tag{3}$$

Constant Vectors

$$\overrightarrow{C_{12}} = \overrightarrow{r_1} \times \overrightarrow{r_2} \tag{4}$$

$$\overrightarrow{C_{23}} = \overrightarrow{r_2} \times \overrightarrow{r_3} \tag{5}$$

$$\overrightarrow{C_{31}} = \overrightarrow{r_3} \times \overrightarrow{r_1} \tag{6}$$

$$\vec{N} = \boldsymbol{r_1}\overrightarrow{C_{23}} + \boldsymbol{r_2}\overrightarrow{C_{31}} + \boldsymbol{r_3}\overrightarrow{C_{12}} \tag{7}$$

$$\vec{D} = \overrightarrow{C_{12}} + \overrightarrow{C_{23}} + \overrightarrow{C_{31}} \tag{8}$$

$$\vec{S} = \overrightarrow{r_1}(\boldsymbol{r_2} - \boldsymbol{r_3}) + \overrightarrow{r_2}(\boldsymbol{r_3} - \boldsymbol{r_1}) + \overrightarrow{r_3}(\boldsymbol{r_1} - \boldsymbol{r_2}) \tag{9}$$

Velocity Vector using three Position Vectors

$$\vec{V} = \left(\frac{\mu}{\boldsymbol{N} \cdot \boldsymbol{D}}\right)^{\frac{1}{2}} \{\vec{D} \times \hat{r} + \vec{S}\} \tag{10}$$

Vis-Viva Equation

$$V^2 = \mu\left(\frac{2}{r} - \frac{1}{a}\right) \tag{11}$$

Appendix C

Extended Kalman Filter

Dynamics

Predicted State

$$\hat{\chi}_k = A\vec{\chi}_{k-1} + \vec{v}_{k-1}$$

State Propagator

$$\vec{\chi}_{k-1} = \begin{bmatrix} \chi_{k-1} \\ y_{k-1} \end{bmatrix}$$

Dynamics Matrix of a Circle

$$A = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix}$$

$$\theta = 0.37\ deg/s$$

Measurement

Measurement Prediction

$$\hat{y}_k = H_k\hat{\chi}_k + \vec{w}_k$$

Jacobian of the Observation Model

$$h(\hat{\chi}_k) = \sqrt{\hat{\chi}_k{}^2 + \hat{y}_k{}^2}$$

$$H_k = \begin{bmatrix} \dfrac{\hat{\chi}_k}{\sqrt{\hat{\chi}_k{}^2 + \hat{y}_k{}^2}} & \dfrac{\hat{y}_k}{\sqrt{\hat{\chi}_k{}^2 + \hat{y}_k{}^2}} \\ \dfrac{-\hat{y}_k}{\hat{\chi}_k{}^2 + \hat{y}_k{}^2} & \dfrac{\hat{\chi}_k}{\hat{\chi}_k{}^2 + \hat{y}_k{}^2} \end{bmatrix}$$

Covariance Matrices and Gain

Dynamic noise covariance

$$\vec{v}_k \sim \mathcal{N}(0, Q)$$

Measurement noise covariance

$$\vec{w}_k \sim \mathcal{N}(0, R)$$

Kalman Gain

$$K_k = \hat{P}_k H_k \left[H_k \hat{P}_k H_k{}^T + R\right]^{-1}$$

Posterior Update

State

$$\vec{\chi}_k = \hat{\chi}_k + K_k\left(\hat{y}_k - h(\vec{\chi}_k)\right)$$

Covariance

$$P_k = (I - K_k H_k)\hat{P}_k$$

REFERENCES

[1] https://www.nasa.gov/audience/forstudents/k-4/stories/nasa-knows/what-is-orbital-debris-k4.html

[2] https://www.nasa.gov/mission_pages/station/news/orbital_debris.html

[3] http://www.esa.int/Our_Activities/Space_Engineering_Technology/Clean_Space

[4] National Research Council. Orbital debris: A technical assessment. National Academies Press, 1995.

[5] National Aeronautics and Space Administration, "Satellite Collision Leaves Significant Debris Clouds," Orbital Debris Quarterly News, vol. 13, no. 2, Apr-2009.

[6] http://www.spacefoundation.orglprograms/public-pol icy-andgovernmentaffairs/introduction-space-activities/space-situational

[7] http://www.onorbitwatch.com/feature/space-situational-awareness-topicdeservesdiscussion

[8] J. A. Kennewell and B.-N. Vo, "An Overview of Space Situational Awareness," 16th International Conference on Information Fusion, Jul. 2013.

[9] https://m.esa.int/Our_Activities/Operations/Space_Situational_Awareness/About_SSA

[10] G. H. McCall and J. H. Darrah, "Space Situational Awareness: Difficult, Expensive--and Necessary," Air and Space Power Journal, pp. 6–16, 2014.

[11] https://www.mnn.com/earth-matters/space/stories/iss-may-shoot-space-trash-out-of-the-sky

[12] http://spacenews.com/editorial-the-breakup-of-dmsp-f13/

[13] R. I. Abbot and T. P. Wallace, "Decision Support in Space Situational Awareness," Lincoln Laboratory Journal, vol. 16, no. 2, pp. 297–335, 2007.

[14] Jones, Brandon A., et al. "Challenges of multi-target tracking for space situational awareness." Information Fusion (FUSION), 2015 18th International Conference on. IEEE, 2015

[15] Committee for the Assessment of the U.S. Air Force's Astrodynamic Standards; Aeronautics and Space Engineering Board; Division on Engineering and Physical Sciences; National Research Council, Continuing Kepler's Quest: Assessing Air Force Space Command's Astrodynamics Standards. The National Academies Press, 2012.

[16] https://www.orbitaldebris.jsc.nasa.gov/photo-gallery.html

[17] https://sciencesprings.wordpress.com/tag/noao/

[18] Hjelmare, Fredrik, and Jonas Rangsjö. "Simultaneous localization and mapping using a Kinect in a sparse feature indoor environment." (2012).

[19] https://www.mathworks.com/hardware-support/turtlebot.html

[20] https://levelskip.com/consoles/How-to-Troubleshoot-Red-Light-on-Xbox-Kinect

[21] http://www.ros.org/

[22] Thavasi, P. Thanu, and C. D. Suriyakala. "Sensors and Tracking Methods Used in Wireless Sensor Network Based Unmanned Search and Rescue System-A Review." Procedia engineering 38 (2012): 1935-1945.